UNIVERSITY OF
ILLINOIS LIBRARY
AT URBANA-CHAMPAIGN
ENGINEERING

APR 24 1981

The person charging this material is responsible for its return to the library from which it was withdrawn on or before the **Latest Date** stamped below.

Theft, mutilation, and underlining of books are reasons for disciplinary action and may result in dismissal from the University.

UNIVERSITY OF ILLINOIS LIBRARY AT URBANA-CHAMPAIGN

ENGINEERING

PHOTO REPRODUCTION
APR 08 REC'D
INTERLIBRARY LOAN

PHOTO REPRODUCTION
APR 13 REC'D
INTERLIBRARY LOAN

MAY 15 REC'D

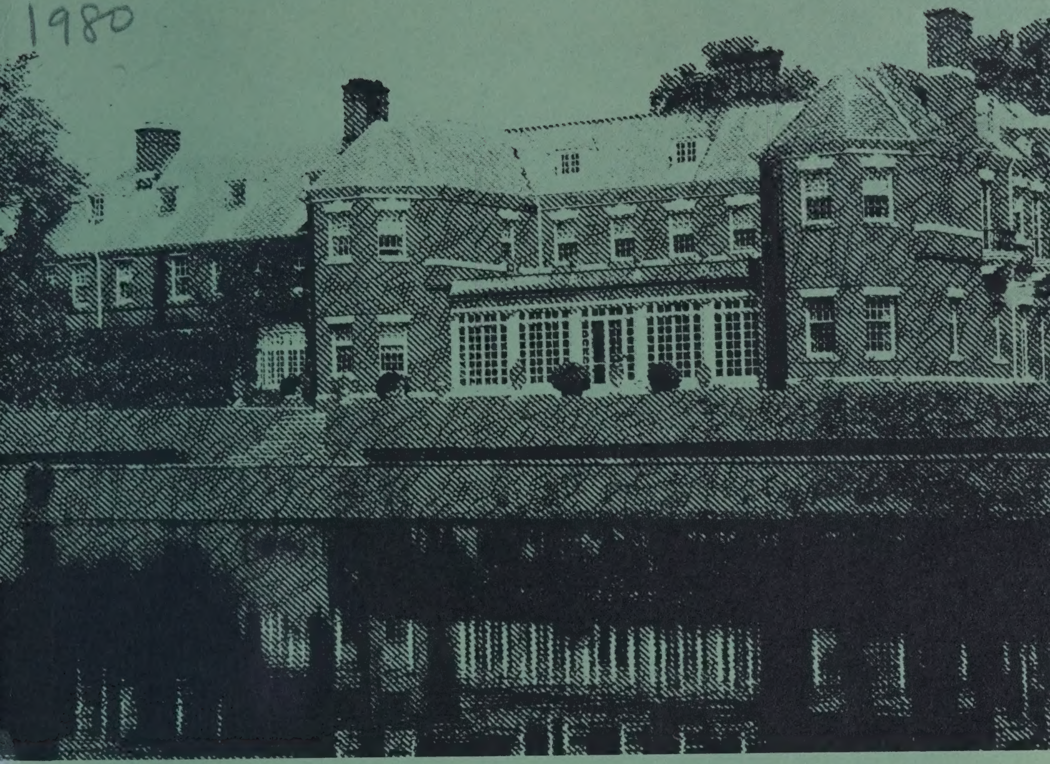
DEC 9 1988
REC'D NOV 14 '88

IRLC

NOV 02 REC'D

L161—O-1096

621.3811
Al5
1980
Engin.



Proceedings

EIGHTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

October 8-10, 1980

Allerton House, Monticello, Illinois
Sponsored by the
Department of Electrical Engineering and the
Coordinated Science Laboratory of the
University of Illinois at Urbana-Champaign

PROCEEDINGS

EIGHTEENTH ANNUAL ALLERTON CONFERENCE ON COMMUNICATION, CONTROL, AND COMPUTING

D. V. Sarwate
W. R. Perkins
Conference Co-Chairmen

Conference held
October 8-10, 1980
Allerton House
Monticello, Illinois

Sponsored by
The Department of Electrical Engineering
and
The Coordinated Science Laboratory
of the
UNIVERSITY OF ILLINOIS
at
Urbana-Champaign

FOREWORD

The Eighteenth Annual Allerton Conference on Communication, Control and Computing was held at Allerton House, Monticello, Illinois on October 8-10, 1980. The conference was sponsored by the Department of Electrical Engineering and the Coordinated Science Laboratory of the University of Illinois at Urbana-Champaign. These PROCEEDINGS contain the papers presented at the conference.

In addition to the large numbers of contributed papers, there were three organized sessions of invited papers and a plenary address. We are grateful to Dr. J. H. Rumbaugh of the U. S. Department of Energy for organizing and chairing a session on Ocean Thermal Energy Conversion and Central Energy Storage, and one on New Source Technology Integration Issues, to Professor R. J. Thomas of Cornell University for organizing and chairing a session on Power Systems Stability, and to Dr. Stephen Kahne of the National Science Foundation for delivering the plenary address titled "NSF Support of Basic Electrical Engineering Research." Their time and efforts on behalf of the Conference are deeply appreciated. We would also like to express our gratitude to the other session chairmen and authors for their contributions to the success of the Conference. The musical contributions of The Soft Hexachord to the social program are also gratefully acknowledged.

The support and encouragement of Professor G. W. Swenson, Jr., Head of the Department of Electrical Engineering, and Professor R. T. Chien, Director of the Coordinated Science Laboratory are appreciated and acknowledged. We would also like to thank our colleagues, Professors J. A. Abraham, N. Ahuja, D. J. Brown, J. B. Cruz, Jr., B. E. Hajek, W. K. Jenkins, P. V. Kokotovic, M. R. Lightner, D. P. Looze, R. J. McEliece, D. E. Muller, D. C. Munson, Jr., J. H. Patel, H. V. Poor, F. P. Preparata, M. B. Pursley, P. W. Sauer, M. S. Schlansker, S. Swamy, T. N. Trick, M. E. Van Valkenburg, and N. Wax for their generous assistance in organizing the Conference.

We are grateful to Mr. Harold Lawler for his handling of the business affairs of the Conference. Finally, very special thanks are due to Ms. Rose Harris and Mrs. Phyllis Young for their gracious and impeccable assistance in all aspects of the organization of the Conference, from the initial preparation of the Conference announcement through Conference registration to the final editing of these PROCEEDINGS.

D. V. Sarwate

W. R. Perkins

TABLE OF CONTENTS

FOREWORD iii

I-A: ALGORITHMICS I

Chairman: F. P. Preparata
University of Illinois

EXPLOITING LINEAR MERGING AND EXTRA STORAGE IN THE MAINTENANCE OF FULLY DYNAMIC GEOMETRIC DATA STRUCTURES 1
I. G. Gowda and D. G. Kirkpatrick

ALGORITHMS FOR TESTING THE INCLUSION OF POINTS IN POLYGONS 11
J. L. Bentley, W. Carruthers

DECOMPOSING A SIMPLE POLYGON WITH THE RELATIVE NEIGHBORHOOD GRAPH 20
G. T. Toussaint

PERMUTATION INVERSIONS AND MULTIDIMENSIONAL CUMULATIVE DISTRIBUTION FUNCTIONS 29
J. M. Hart

LOWER BOUNDS FOR GEOMETRIC PROBLEMS 35
D. Avis

AN ANALYSIS OF TWO HEURISTICS FOR THE EUCLIDEAN TRAVELING SALESMAN PROBLEM 41
J. L. Bentley and J. B. Saxe

I-B: CODING AND SIGNAL DESIGN

Chairman: R. J. McEliece
University of Illinois

ON EXTENDED BCH CODES 50
R. E. Blahut

REDUCED-STATE SOFT-DECISION TRELLIS DECODING OF LINEAR BLOCK CODES 60
K. R. Matis and J. W. Modestino

VORONOI REGIONS, SPHERE PACKINGS AND QUANTIZATION 67
N. J. A. Sloane

INDEPENDENCE OF ERROR EVENT WAITING TIMES IN CONVOLUTIONAL DECODERS 68
K. Srivatsa and J. B. Anderson

DESIGN OF SIGNAL SETS WITH SPECIFIED CORRELATION PROPERTIES 78
T. P. McGree and G. R. Cooper

EFFICIENT TRANSMISSION OF A SET OF SIGNALS OVER FEWER CHANNELS BY LINEAR TRANSFORMATIONS AND TIME SHIFTING 88
N. Kalouptsidis, C. Caroubalos, D. Manolakis

MODULATOR DESIGN FOR A BANDLIMITED MULTIPLE ACCESS CHANNEL 89
P. Narayan and D. L. Snyder

I-C: LINEAR CONTROL SYSTEMS

Chairman: K. D. Young
Drexel University

PENETRATING TRANSMISSION ZEROS IN THE DESIGN OF ROBUST SERVOMECHANISM SYSTEMS 91
S. H. Wang and E. J. Davison

ON A DISTURBANCE COMPENSATION METHOD 98
J. Kim and D. H. Chyung

DYNAMIC INVERSE OF GENERALIZED STATE SPACE SYSTEMS 104
V. A. Oliveira and S. P. Bhattacharyya

POLE ASSIGNMENT COMPENSATORS WITH ARBITRARY SPECTRUM 105
S. P. Bhattacharyya

A NEW RESULT FOR OUTPUT FEEDBACK DECOUPLING 106
H. Y. Kim and E. Y. Shapiro

ANALYSIS OF SINGULAR SINGULARLY-DISTURBED SYSTEMS 116
K. D. Young

DIGITALIZATION OF THE CONTINUOUS-DATA SYSTEMS VIA TIME-RESPONSE MATCHING 124
C.-G. Chung, K.-W. Han, and H.-H. Yeh

I-D: OCEAN THERMAL ENERGY CONVERSION AND CENTRAL ENERGY STORAGE

Organizer and Chairman: J. H. Rumbaugh
U. S. Department of Energy

INTEGRATING OTEC INTO POWER SYSTEM PLANNING STUDIES 134
G. E. Marks

DYNAMIC INTERACTION BETWEEN AN OTEC POWER PLANT AND A POWER GRID 145
A. Lavi

OTEC DYNAMIC SIMULATION MODELS 155
D. N. Wormley and G. Thomas

POTOMAC ELECTRIC POWER COMPANY COMPRESSED AIR ENERGY STORAGE STUDY 167
P. E. Schaub and E. D. Shippey

POTOMAC ELECTRIC POWER COMPANY UNDERGROUND PUMPED HYDRO STUDY 177
P. E. Schaub and F. Pelton

II-A: NEW SOURCE TECHNOLOGY INTEGRATION ISSUES

Organizer and Chairman: J. H. Rumbaugh
U. S. Department of Energy

SIMULATION AND OPERATIONS OF LARGE WIND MACHINES 187
O. Wasynczuk

AN IMPACT ASSESSMENT OF MOD-2 WIND GENERATORS ON THE TVA POWER SYSTEM 196
D. W. Hilson, N. D. Sadanandan, H. M. Sendaula, and R. R. Johnson

INTEGRATION PROBLEMS WITH LARGE WIND ENERGY CONVERSION SYSTEMS 206
R. J. Thomas and J. S. Thorp

POWER SYSTEM PLANNING AND THE NEW TECHNOLOGIES 214
R. L. Sullivan

II-B: ALGORITHMICS II - COMBINATORIAL COMPUTING

Chairman: C. L. Liu
University of Illinois

DYNAMIC DATA STRUCTURES: FINITE FILES, LIMITING PROFILES, AND VARIANCE ANALYSIS 223
L. Cheno, P. Flajolet, J. Francon, C. Puech, and J. Vuillemin

AVERAGE PERFORMANCE OF THE B-TREE 233
W. E. Wright

AN UPPER BOUND FOR SORTING INTEGERS IN RESTRICTED RANGES 243
D. Kirkpatrick

HYBRID SORTING ALGORITHMS 250
S. G. Akl and H. Meijer

EXTERNAL SORTING REVISITED: APPLICATION OF DISTRIBUTIVE METHODS 260
D. J. Hogan and B. W. Weide

AN ON-CHIP SORTER/MERGER 267
D. T. Lee, H. Chang, and C. K. Wong

II-C: IDENTIFICATION AND MODELING

Chairman: A. H. Haddad
National Science Foundation

IDENTIFICATION OF MULTIVARIABLE LINEAR SYSTEMS 275
J. Wang

ROUTH-HURWITZ APPROXIMATION OF MULTIVARIABLE SYSTEMS 285
N. K. Sinha, I. El-Nahas and R. T. H. Alden

TRANSFER FUNCTION IDENTIFICATION FOR MULTI-VARIABLE LINEAR SYSTEMS 295
G.-H. Xia

ON-LINE PARTIAL REALIZATIONS OF A DISCRETE-TIME NON-LINEAR BLACK BOX BY LINEAR REPRESENTATION SYSTEMS 303
T. Matsuo

IDENTIFICATION AND MODEL APPROXIMATION FOR CONTINUOUS-TIME SYSTEMS ON FINITE PARAMETER SETS 313
J. K. Tugnait

PARTIAL REALIZATIONS OF LINEAR STOCHASTIC TIME-VARYING SYSTEMS 323
R. E. Skelton

II-D: COMMUNICATION NETWORKS

Chairman: B. E. Hajek
University of Illinois

MODELING AND PERFORMANCE ANALYSIS OF THE PACKET FORMATION PROCESS IN PACKET-SWITCHED TELECOMMUNICATIONS 333
J. S. Meditch

"REGULARITY" AND DELAY IN QUEUEING SYSTEMS 339
A. Ephremides

A QUEUEING MODEL FOR ANALYSIS OF A BURSTY MULTIPLE-ACCESS COMMUNICATION CHANNEL 340
J. W. Sennott and L. I. Sennott

ON DESIGN OF THE LEAST VULNERABLE NETWORKS OF MINIMUM DELAY 341
H. Lee

II-E: COMMUNICATION SYSTEMS
Chairman: J. W. Modestino
Rensselaer Polytechnic Institute

PERFORMANCE SIMULATION OF RECEIVER FILTERS FOR SERIAL DETECTION OF MSK SIGNALS 351
J. M. Liebetreu and C. R. Ryan

AVERAGE ERROR PROBABILITY FOR DIRECT SEQUENCE SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEMS 359
K.-T. Wu and D. L. Neuhoff

PROGRESS IN DIGITAL MODULATION TECHNIQUES FOR SATELLITE SYSTEMS 369
W. W. Wu and T. R. Rowbotham

A CHANNEL MODEL FOR BLOCKED DATA COMMUNICATION 379
P. K. Varshney

PLENARY SESSION

NSF SUPPORT OF BASIC ELECTRICAL ENGINEERING RESEARCH
Dr. Stephen Kahne

III-A: ROBUSTNESS AND SENSITIVITY
Chairman: D. P. Looze
University of Illinois

UNSTABLE SYSTEMS AND FEEDBACK: THE GAP METRIC 380
G. Zames and A. El-Sakkary

AN APPLICATION OF TOTAL SYNTHESIS TO ROBUST COUPLED DESIGN 386
M. K. Sain, R. M. Schafer, and K. P. Dudek

GAIN AND PHASE MARGINS FOR LINEAR MULTIVARIABLE FEEDBACK SYSTEMS..................... 396
I. Postlethwaite

ON THE ROBUSTNESS OF OUTPUT FEEDBACK CONTROL METHODS TO MODELING ERRORS 405
H. K. Khalil

CONTROLLER DESIGN FOR PARAMETER SENSITIVITY REDUCTION IN LINEAR REGULATORS 414
R. Yedavalli and R. E. Skelton

ON SENSITIVITY OF NONLINEAR DIGITAL SYSTEMS 424
M. Eslami

III-B: SOURCE CODING AND QUANTIZATION
Chairman: J. B. Anderson
McMaster University

QUANTIZATION-PERMUTATION CODES AND BUFFER-ADAPTED HUFFMAN CODES 433
T. Berger, M. U. Chang, and S.-Y. Tung-Kleinberg

TREE-SEARCHED BLOCK SOURCE CODES 437
R. M. Gray

VARIABLE-LENGTH TO VARIABLE-LENGTH ENCODERS ARE ASYMPTOTICALLY MEAN STATIONARY 438
J. C. Kieffer and J. G. Dunham

THE PRINCIPLE OF CONSERVATION OF ENTROPY 440
J. G. Dunham

THE DESIGN OF MULTIDIMENSIONAL QUANTIZERS USING PREQUANTIZATION 446
K. D. Rines and N. C. Gallagher, Jr.

A NOTE ON MULTIDIMENSIONAL ASYMPTOTIC QUANTIZATION THEORY 454
J. A. Bucklew and G. L. Wise

III-C: ALGORITHMICS III - SYSTEMS
Chairman: E. M. Reingold
University of Illinois

ON THE COMPLEXITY OF FIXED-PRIORITY SCHEDULING OF REAL-TIME TASKS 464
J. Y.-T. Leung and J. Whitehead

ON SCHEDULING INDEPENDENT TASKS WITH RESTRICTED EXECUTION TIMES 471
J. Y.-T. Leung

OPTIMAL TWO PROCESSOR SCHEDULING OF TREE PRECEDENCE CONSTRAINED TASKS WITH TWO EXECUTION TIMES 480
K. Nakajima, J. Y.-T. Leung, and S. L. Hakimi

TESTING LOOP-FREE CONTROL STRUCTURES 490
E. Leiss and F. D. Lewis

AUTHORIZATION SYSTEMS WITH BOUNDED PROPAGATION 491
E. Leiss

MARKED GRAPH TRANSFORMATION BY STEPWISE TRANSITION REFINEMENTS 497
I. Suzuki and T. Murata

SUBGRAPH REACHABILITY OF MARKED GRAPHS 507
Y. Kajitani and T. Kawakami

ON THE NODE-NUMBERING PROBLEM 514
S. R. Soloway

III-D: COMPUTER SYSTEMS
Chairman: M. S. Schlansker
University of Illinois

A BINOMIAL MODEL FOR SOFTWARE PERFORMANCE PREDICTION 524
J. G. Shanthikumar

PRELIMINARY DESIGN OF A HIGHLY PARALLEL ARCHITECTURE FOR REAL-TIME APPLICATIONS 534
M. E. Brown and B. W. Weide

PERMUTING WITH THE AUGMENTED DATA MANIPULATOR NETWORK 544
R. J. McMillen, G. B. Adams, III, and H. J. Siegel

MODELING OF SYNCHRONIZED ITERATIVE ALGORITHMS FOR MULTIPROCESSORS 554
F. A. Briggs and M. Dubois

III-E: ALGEBRAIC COMPLEXITY
Chairman: A. Sameh
University of Illinois

A JACOBI ALGORITHM AND ITS IMPLEMENTATION ON PARALLEL COMPUTERS 564
K-W. Chen and K. B. Irani

EFFICIENT INVERSION FORMULAS FOR SUMS OF PRODUCTS OF TOEPLITZ AND HANKEL MATRICES 574
B. Friedlander and M. Morf

IV-A: NONLINEAR SYSTEMS
Chairman: N. Wax
University of Illinois

VARIABLE METRIC METHODS FOR CONSTRAINED OPTIMIZATION USING DIFFERENTIABLE EXACT PENALTY FUNCTIONS 584
D. P. Bertsekas

CONTROL AND COMPUTING FOR AUTOMATED INTENSIVE CARE OF THE CARDIOVASCULAR SYSTEM 594
N. DeClaris and K. M. Kempner

GENERAL COMPOSITE CONTROL LAWS FOR NONLINEAR SINGULARLY PERTURBED SYSTEMS 595
M. Suzuki

A TENSOR APPROACH TO MODELING OF NONHOMOGENEOUS NONLINEAR SYSTEMS 604
S. Yurkovich and M. Sain

IV-B: ADAPTIVE CONTROL
Chairman: J. B. Cruz, Jr.
University of Illinois

THE SIMPLEST BAYESIAN ADAPTIVE CONTROL PROBLEM 614
P. R. Kumar

OPTIMAL ADAPTIVE CONTROLLERS FOR FINITE STATE SYSTEMS 618
P. R. Kumar and A. Becker

THE GLOBAL CONVERGENCE OF DIRECT ADAPTIVE INPUT MATCHING CONTROL OF SOME NONMINIMUM PHASE PLANTS 623
C. R. Johnson, Jr., G. C. Goodwin and K. S. Sin

THE USE OF ADAPTIVE CONTROL FOR HELICOPTER TRAJECTORIES IN SEARCH OPERATIONS 633
A. Chacón, R. Padilla, C. Padilla, M. Azagoury, and B. Fernández

IV-C: ALGORITHMICS IV - GRAPHS AND NETWORKS
Chairman: D. E. Muller
University of Illinois

K-COLORING OF BIPARTITE GRAPHS 643
V. Ashok and J. Ja'Ja'

BANDWIDTH PROBLEMS IN GRAPHS 650
B. Monien and I. H. Sudborough

ANALYSIS OF A CLASS OF GRAPH PARTITIONING PROBLEMS 660
P. Bertolazzi, M. Lucertini, and A. M. Spaccamela

ALGORITHMS FOR FINDING PATHS WITH MULTIPLE CONSTRAINTS 668
J. M. Jaffe

ON FINDING SEVERAL SHORTEST PATHS IN CERTAIN GRAPHS 677
Z. M. Kedem and H. Fuchs

EFFICIENT DISTRIBUTED ALGORITHMS FOR FINDING CENTERS AND MEDIANS IN NETWORKS 687
E. Korach, D. Rotem and N. Santoro

IV-D: POWER SYSTEMS STABILITY
Organizer and Chairman: R. J. Thomas
Cornell University

OPRIMAL QUADRATIC STABILIZABILITY OF UNCERTAIN LINEAR SYSTEMS 697
C. V. Hollot and B. R. Barmish

STABILITY OF SELF PROTECTING SYSTEMS 707
J. S. Thorp, R. J. Thomas, and J. W. McClure

TRADEOFFS IN THE DESIGN OF MULTIVARIABLE FEEDBACK SYSTEMS 715
J. E. Wall, Jr., J. C. Doyle, and C. A. Harvey

GENERALIZED DECOMPOSITIONS AND STABILITY OF NONLINEAR SYSTEMS 726
M. Ikeda and D. D. Siljak

DIRECT METHODS FOR OFF-LINE AND ON-LINE TRANSIENT STABILITY ASSESSMENT OF POWER SYSTEMS 735
M. Ribbens-Pavella, J. Sabatel, A. Calvaer, and J. Gheury

IV-E: SIGNAL DETECTION AND ESTIMATION
Chairman: H. V. Poor
University of Illinois

SIGNAL DETECTION BASED ON DISCRETE DATA 745
C. C. Lee and J. B. Thomas

THE PERFORMANCE CHARACTERISTICS OF TWO EXTENSIONS OF THE SIGN DETECTOR 755
S. A. Kassam

ON THE PERFORMANCE OF A MODIFIED SIGN DETECTOR FOR STRONG MIXING NOISE 765
D. R. Halverson and G. L. Wise

ROBUST DETECTION OF KNOWN SIGNALS IN ASYMMETRIC NOISE 774
S. A. Kassam, G. Moustakides, and J. G. Shin

PERFORMANCE ANALYSIS OF AN OPTIMUM DETECTOR WITH COUNTING POINT PROCESS OBSERVATIONS 784
A. A. Lazar and S. C. Schwartz

A NEW CONCEPT OF MULTIPLE-PARAMETER QUANTUM MECHANICAL ESTIMATION WITH APPLICATIONS IN OPTICAL COMMUNICATIONS 792
D. D. Ilic

A NEW ROBUST ESTIMATOR OF LOCATION PARAMETER 801
C. A. Pomalaza and C. D. McGillem

V-A: ALGORITHMICS V - PARALLEL COMPUTATION
Chairperson: D. J. Brown
University of Illinois

EXPERIMENTS WITH A PARALLEL COMMUNICATION SCHEME 802
L. G. Valiant

A PROCESSOR-EFFICIENT CONNECTED COMPONENTS ALGORITHM 812
C. M. Fiduccia and R. M. Mattheyses

LINEAR TRANSFORMATION PERMUTATIONS IN MULTISTAGE INTERCONNECTION NETWORKS 822
M. A. Abidi and J. J. Metzner

ELECTION PROCESSES IN DISTRIBUTED SYSTEMS 823
D. S. Hirschberg

ON PARALLEL SEARCHING OF FILES IN A DISTRIBUTED COMPUTING SYSTEM 824
S. L. Dallas and S. L. Hakimi

MUTUAL EXCLUSION USING INDIVISIBLE READS AND WRITES 833
J. E. Burns and N. A. Lynch

IMPROVING ACCESS INTERFERENCE IN DISTRIBUTED DATA BASES 843
G. Martella and F. Schreiber

V-B: LARGE SCALE SYSTEMS
Chairman: P. V. Kokotovic
University of Illinois

ALGORITHMS FOR LEADER-FOLLOWER GAMES 851
G. P. Papavassilopoulos

ON DUALITY IN STOCHASTIC GAMES 860
R. A. Milito, R. A. Padilla and C. S. Padilla

PRODUCT FORM AND OPTIMAL LOCAL FEEDBACK FOR A MULTIINDEX MARKOV CHAIN 870
J. P. Quadrat and M. Viot

NESTED OPTIMIZATION OF WEAKLY COUPLED MARKOV CHAINS 881
S. H. Javid

A NEW ALGORITHM FOR NONDIFFERENTIABLE OPTIMIZATION AND ITS USE FOR COMPUTING SADDLE POINTS 891
G. Cohen and J.-F. Balducchi

ROBUST DECENTRALIZED CONTROL USING OUTPUT FEEDBACK 900
Ö. Hüseyin, M. E. Sezer, and D. D. Siljak

EXPLOITING LINEAR MERGING AND EXTRA STORAGE IN THE MAINTENANCE OF FULLY DYNAMIC GEOMETRIC DATA STRUCTURES

IHOR G. GOWDA and DAVID G. KIRKPATRICK
Department of Computer Science, University of British Columbia,
Vancouver, B.C., Canada V6T 1W5

1. INTRODUCTION AND OVERVIEW

1.1 Maintenance of Dynamic Data Structures

The bulk of early work in computational geometry has been concerned with what are essentially static problems. Consequently the majority of data structures have been designed with little or no concern for the efficient insertion and deletion of objects (eg. points). In many cases a complete reconstruction of the data structure is required to reflect the insertion or deletion of even a single object.

Bentley [2] has recently observed that, for some problems, a distributed approach can provide reasonably efficient alternatives to the familiar static data structures. Rather than maintaining an entire set of objects in one "global" data structure, it may be advantageous to partition the set into smaller statically organized pieces, each of which is maintained separately. Bentley identifies a large class of problems (called decomposable searching problems) for which this general technique of converting static data structures into dynamic ones ("dynamization") is applicable.

Bentley and Saxe [2,13] have presented several general dynamization techniques which can be applied to decomposable searching problems. Their techniques are designed primarily to accomodate insertions (as opposed to deletions which are much more difficult to handle in their framework). More recently, a number of techniques have been proposed [10,11,16] for dealing with insertions and deletions more uniformly. The results of this paper continue this trend. A general technique is presented by which deletion costs can be reduced at the expense of extra storage. This technique is compatible with but does not depend on the notion of decomposability.

1.2 Preliminary Definitions and Notation

We will be interested in two types of data structures and their relationships. A static structure S is built once and can be searched (or otherwise interrogated) repeatedly, but is not meant to be updated. We define three functions of n (the size or number of objects represented by S) which characterize the behaviour of S:

$P_S(n)$ = the preprocessing time required to build S,

$Q_S(n)$ = the time required to perform a single query in S,

and $S_S(n)$ = the space required to represent S.

We restrict our attention to worst case behaviour for these (and subsequent) attributes of structures.

A dynamic structure represents a set of objects that changes over time. That is, it must support the insertion and deletion of new objects, in addition to queries. The following are used to characterize the performance of a dynamic structure D representing a set of size n:

$I_D(n)$ = the time required to insert an element into D,

$D_D(n)$ = the time required to delete an element from D,

$Q_D(n)$ = the time required to perform a single query in D,

$S_D(n)$ = the space required to represent D,

and $P_D(n)$ = the preprocessing time required to build D.

1.3 Overview

In section 2, we illustrate our dynamization technique with specific reference to Voronoi diagrams. A particularly important application to dynamic nearest neighbour searching is developed in Section 3. Section 4 describes our technique in more general terms, setting the stage for a generalization that is developed and illustrated in Section 5.

2. DYNAMIC VORONOI DIAGRAMS

2.1 Motivation and Background

Given a set S of points in the plane, we can associate with each point $p \in S$ a region of the plane (its Voronoi polygon) formed by points closer to p than to any other element of S. This partition of the plane (equivalently the collection of edges separating the Voronoi polygons) is called the Voronoi diagram of S. Voronoi diagrams are among the most versatile of geometric data structures [14,15], and they have given rise to a number of interesting algorithmic issues and generalizations [8,15]. (We will recall a number of familiar applications of Voronoi diagrams in section 2.3).

The efficient construction of Voronoi diagrams is based on the following:

Lemma 2.1 [15] If X and Y are linearly separable planar point sets then the Voronoi diagram of $X \cup Y$ can be constructed from the Voronoi diagram of X and the Voronoi diagram of Y in $O(|X| + |Y|)$ steps.

Lemma 2.1 leads to an $O(n \lg n)$ algorithm [15] for the construction of Voronoi diagrams on point sets of size n, which is known to be optimal for a comparison based model of computation [15]. While it does not lead to an asymptotically improved algorithm, it is interesting to note that the separability condition of lemma 2.1 can be dropped. Specifically,

Lemma 2.2 [8] If X and Y are arbitary planar point sets then the Voronoi diagram of $X \cup Y$ can be formed from the Voronoi diagram of X and the Voronoi diagram of Y in $O(|X| + |Y|)$ steps.

An immediate corollary of lemma 2.2 is that a Voronoi diagram can be updated, following the insertion of a new data point, in linear time. It also shows that two Voronoi diagrams can be merged together faster than the resulting diagram could be built "from scratch"; this observation will be discussed further in Section 3. It is natural to ask if it is possible to perform point deletions from a Voronoi diagram faster than the time required to reconstruct the entire Voronoi diagram on the updated point set.

2.2 Exploiting Extra Storage for Faster Deletion

It turns out that, by using extra storage, we can exploit lemma 2.2 to achieve a speedup in the cost of deletion from a Voronoi diagram. Specifi-

621.3811
Al5
1980

engineer

cally,

Theorem 2.3 Voronoi diagrams of size n can be dynamically maintained in O(n) time per insertion or deletion.

Proof. A Voronoi diagram D will be maintained at the root of a balanced binary tree T_D of Voronoi diagrams. The leaves of T_D are one point Voronoi diagrams and each internal node has associated with it a Voronoi diagram formed by merging the Voronoi diagrams of its two sons. (Since, in practice, Voronoi diagrams are usually constructed by merging smaller Voronoi diagrams, we can think of T_D as being a "history" of the construction of D).

The tree structure T_D can be maintained dynamically in time proportional to the size of D, per update.

Insertion: A new point is added as a new leaf of T_D and, if necessary, T_D is rebalanced. The Voronoi diagrams of T_D must be updated along the entire path from the new leaf to the root (as well as wherever rebalancing has assigned new sons to a node). The dominant cost of an insertion will be the cost of updating the root diagram (from its updated sons). That is, insertion into and maintenance of T_D costs no more (asymptotically) than insertion directly into D.

Deletion: The leaf point to be deleted is found (in O(lgn) time using an auxiliary dictionary) and removed, and T_D is updated along the path from this leaf to the root. As before, since even with restructuring at most a constant number of Voronoi diagrams need be updated at each level, the entire deletion process can be accomplished in time linear in the size of D. □

As we have described it, if D is a Voronoi diagram on n points then T_D uses O(nlgn) storage (each point is represented on each level of the tree). In fact the same result is achieved if T_D is maintained only to a depth of lg lg n (that is, the leaves are Voronoi diagrams on n/lgn points), for a total of n lg lg n storage. We maintain O(lgn) leaf sets each of size O(n/lgn) using the set balancing technique of van Leuwen and Wood [16]:

Lemma 2.4 [16] If k(n) is monotonic increasing, it is possible to dynamically maintain a set of n points in O(k(n)) subsets each of size O(n/k(n)) at a cost of O(lgn) steps per update.

Insertions and deletions proceed as before except that a leaf structure may have to be completely reconstructed with each update (recall that insertion can be done more cheaply). The cost of leaf reconstruction is O((n/lgn)lg(n/lgn)), that is O(n), which matches the cost of updating the remainder of the tree T_D.

In fact, the same idea can be used with even less storage (fewer levels in T_D). Let F(n), $1 \leq F(n) \leq$ lglgn, be the level to which we maintain T_D for Voronoi diagrams D of size n. Then the strategy described above leads to,

Theorem 2.5 Voronoi diagrams of size n can be dynamically maintained in O(n) time per insertion and $O(n\ \mathrm{lgn}/2^{F(n)})$ time per deletion, using O(nF(n)) storage.

2.3 Some Applications

As with (or perhaps, because of) their static counterparts dynamic Voronoi diagrams have numerous applications. Perhaps most interesting is the application of dynamic Voronoi diagrams to dynamic nearest neighbour searching. This is discussed in detail in Section 3. Other applications stem directly from the fact that certain geometric structures or properties can be computed in linear time from a Voronoi diagram. This linear factor is simply absorbed in the cost of dynamically maintaining the underlying Voronoi diagram.

As a straightforward example, consider the all-nearest-neighbours problem (given n points in the plane find the nearest neighbour of each among points in the set). Shamos [14] has shown that, given the Voronoi diagram of the point set S, the all-nearest-neighbours problem for S can be solved in $O(|S|)$ time. It follows that,

Application 2.6 The all-nearest-neighbours pairs of a set of n points in the plane can be maintained dynamically at a cost of $O(n)$ steps per insertion or deletion of a point.

In a similar fashion we have the following :

Application 2.7 The Delaunay triangulation (cf [14]) of a set of n points in the plane can be maintained dynamically at a cost of $O(n)$ steps per insertion or deletion of a point.

Application 2.8 The Euclidean minimum spanning tree (cf [14]) of a set of n points in the plane can be maintained dynamically at a cost of $O(n)$ steps per insertion or deletion of a point.

Application 2.9 The largest empty circle defined by n points in the plane (cf [14]) can be maintained dynamically at a cost of $O(n)$ per insertion or deletion of a point.

The above applications are made possible by the following: the Delaunay triangulation of a point set can be constructed from the Voronoi diagram in linear time [14]; the Euclidean minimum spanning tree of a point set can be constructed from the Voronoi diagram in linear time [4,14]; and the largest empty circle defined by a point set can be constructed from the Voronoi diagram in linear time [6].

3. DYNAMIC NEAREST NEIGHBOUR SEARCHING

3.1 Earlier Results

Given a set of data points in the plane, a nearest neighbour query takes the form: which data point is closest to a given test point (usually the Euclidean distance is assumed, but efficient solutions to the problem seem to be reasonably insensitive to the underlying metric). The problem of nearest neighbour searching (also known as the post-office problem) has received considerable attention in both its static and dynamic versions. Since efficient static nearest neighbour search structures are used extensively as components of their dynamic counterparts, it is appropriate to start with a review of the best bounds known for static nearest neighbour searching.

For static nearest neighbour searching, we know that structures exist

[8,9] with the following attributes:

$$Q_S(n) = O(\lg n)$$
$$S_S(n) = O(n)$$
and $$P_S(n) = O(n \lg n).$$

The semi-dynamic version of nearest neighbour searching was discussed by Saxe and Bentley [13] as one of a family of decomposable search problems. Their primary dynamic technique showed that

$$Q_D(n) = O(\lg^2 n)$$
$$S_D(n) = O(n)$$
$$P_D(n) = O(n \lg^2 n)$$
and $$I_D(n) = O(n \lg n)$$

can be achieved. The structure proposed by Saxe and Bentley was modified by Overmars and van Leeuwen [11] to permit deletion, yielding a structure with the same attributes (all updates take $O(n \lg n)$ in the worst case).

Van Leeuwen and Wood [16] (building on results of Maurer and Ottman [10]) exploit lemma 2.4 to achieve the bounds

$$Q_D(n) = O(k \cdot Q_S(n/k))$$
$$U(n) = O(D(n) + U(n/k)),$$

where $U(n)$ is update time (insertion or deletion) and $D(n)$ is the cost of a dictionary check. Using this scheme, they show how to improve the update time to $O(n)$, retaining $Q_D(n) = O(\lg^2 n)$. Edelsbrunner [5] studies the question of optimal tradeoff between query and update times.

3.2 Exploiting Dynamic Voronoi Diagrams

All of the dynamic nearest neighbour searching results described above relate to search structures which, like the best static nearest neighbour searching structures [8,9], are based on the Voronoi diagram(s) of the data set (or subsets of the full data set). As we noted in Section 2.1, Voronoi diagrams have the property that they can be "merged" asymptotically faster than they can be reconstructed. This means (as anticipated in [13]) that the bounds on $I(n)$ and $P(n)$ cited above can, in every case, be reduced by a factor of lgn. This has the effect, for the fully dynamic structures of Overmars and van Leeuwen [11] and van Leeuwen and Wood [16], of making insertion cheaper (by this lgn factor) than deletion. By exploiting the dynamic Voronoi diagrams described in Section 2, we are able to reduce the cost of deletion in dynamic nearest neighbour searching, bringing the costs of insertion and deletion back into balance.

Suppose we replace each Voronoi diagram D in the dynamic nearest neighbour search structure of van Leeuwen and Wood [16] by its dynamic counterpart T_D. Since D is contained in the structure T_D it is easy to see that the query time will be no greater in the resulting structure. Furthermore, updates, which require a total reconstruction of one or more Voronoi diagrams in the van Leeuwen-Wood structure, can now exploit the attributes of our dynamic Voronoi diagrams. It should be evident from this discussion that our technique for reducing deletion cost is fully compatible with the decomposable nature of nearest neighbour searching. This gives rise to two independent tradeoffs, one between query time and update time due to decomposability and the second between deletion time and storage due to the properties of dynamic Voronoi diagrams. Our general result for nearest neighbour searching can be stated as follows:

Theorem 3.1 If $1 \leq F(n) \leq \lg\lg n$ and $1 \leq H(n) \leq n$ then there exists a dynamic nearest neighbour search structure with attributes:

$$Q_D(n) = O(H(n)\lg(n/H(n)))$$

$$S_D(n) = O(nF(n))$$

$$I_D(n) = O(n/H(n) + \lg n)$$

and $$D_D(n) = O\left[\frac{n \lg n}{H(n)2^{F(n)}} + \lg n\right].$$

At the extreme ($F(n) = \lg\lg n$), when we don't care about storage, we have the following,

Corollary 3.2 If $1 \leq H(n) \leq n$, then there exists a dynamic nearest neighbour search structure with attributes:

$$Q_D(n) = O(H(n)\lg(n/H(n)))$$

$$S_D(n) = O(n \lg\lg n)$$

$$I_D(n) = O(n/H(n) + \lg n)$$

and $$D_D(n) = O(n/H(n) + \lg n)$$

If we choose $H(n) \sim (n/\lg n)^{1/2}$ we have a structure with $\max(Q_D(n), I(n), D(n)) = O((n \lg n)^{1/2})$. This should be compared with the lower bound of Bentley and Saxe [13] which shows that $\max(Q_D(n), I(n), D(n)) = \Omega(n^{1/2})$.

4. DYNAMIZATION OF OTHER LINEARLY MERGEABLE STRUCTURES

What properties of Voronoi diagrams allowed us to perform updates faster than the time required for complete reconstruction? Apparently it is the fact that Voronoi diagrams can be merged asymptotically faster than they can be reconstructed (Lemma 2.2). At this stage it is not clear if arbitrary linear merging is necessary; certainly it was exploited in the dynamization procedure described. We shall argue, in the next section, that efficient arbitrary merging is dispensable. However we should note that, without any further complications, our techniques of trading deletion cost for storage in the maintenance of dynamic data structures will apply whenever such structures can be arbitrarily merged faster than they can be entirely reconstructed.

4.1 Dynamic Furthest Point Voronoi Diagrams and Applications

To support our claim regarding the generality of our dynamization technique, consider the problem of dynamic maintenance of furthest point Voronoi diagrams (cf [6]). Shamos and Hoey [15] outline a method for merging the FPVD's of two linearly separated point sets in linear time. A linear algorithm for merging arbitrary FPVD's, similar in spirit to the algorithm of [7] for ordinary Voronoi diagrams, is described in [6]. It follows (by direct analogy with Theorem 2.5) that, when $1 \leq F(n) \leq \lg\lg n$,

Theorem 4.1 Furthest point Voronoi diagrams of size n can be dynamically maintained in $O(n)$ time per insertion and $O(n \lg n/2^{F(n)})$ time per deletion, using $O(nF(n))$ storage.

Like its counterpart for standard Voronoi diagrams, Theorem 4.1 has a number of interesting applications. Specifically,

Theorem 4.2 If $1 \leq F(n) \leq \lg\lg n$ and $1 \leq H(n) \leq n$, then there exists a dynamic furthest-neighbour search structure on n points with attributes

$$Q_D(n) = O(H(n)\lg(n/H(n)))$$

$$S_D(n) = O(nF(n))$$

$$I_D(n) = O(n/H(n) + \lg n)$$

and $$D_D(n) = O\left(\frac{n \lg n}{H(n)2^{F(n)}} + \lg n\right).$$

Proof Analogous to the proof of Theorem 3.1, replacing Theorem 2.5 by Theorem 4.1. □

Other applications exploit the fact that certain attributes of point sets in the plane, specifically the diameter and smallest enclosing circle, can be determined in linear time given the FPVD [15]. Hence,

Application 4.3 The diameter and smallest enclosing circle of a set of points in the plane can be maintained dynamically at a cost of $O(n)$ steps per insertion or deletion of a point.

5. DYNAMIZATION USING RESTRICTED LINEAR MERGING OF STRUCTURES

There are a number of geometric data structures, the convex hull of a set of points in 3-space being a good example, for which we have linear algorithms to merge "separable" structures [12], but know of no such algorithm for the merging of arbitary structures. We are led to ask whether this restricted linear merging can also be exploited to speed up the dynamic maintenance of the associated data structures. It turns out that this restricted linear merging gives rise to speedups of exactly the same order as the unrestricted linear merging discussed earlier.

5.1 Maintaining Linearly Separated Sets of Approximately Equal Size

Theorem 2.3 (maintenance of dynamic Voronoi diagrams), in its full generality, takes advantage of a technique for dynamically maintaining approximately equal sized subsets of a set, described by van Leeuwen and Wood [16]. As it is described this technique does not lend itself to maintaining linearly separable subsets of points. However, it is possible to preserve separability with no (asymptotic) increase in the amount of work. The following should be compared with lemma 2.4.

Lemma 5.1 If $k(n)$ is monotonic increasing, it is possible to dynamically maintain a set of n points in $O(k(n))$ linearly separated subsets each of size $O(n/k(n))$ at a cost of $O(\lg n)$ steps per update.

Proof We shall only sketch the proof here. For details see [6].

The set of n points is maintained in such a way that each subset has a size in the range $[\lfloor n/k(n)\rfloor, 2\lfloor n/k(n)\rfloor + 2]$. If a particular subset lies on either extreme of this range we will call it critical. The following strategy has the property that it maintains the invariant condition: all subset sizes lie inside the specified range and at most $k(n)(\lfloor n/k(n)\rfloor + 1) - n$ subsets are critical.

All subsets are maintained in a priority queue ordered by subset size. On insertion (deletion) a dictionary is consulted to determine membership and to identify the subset which should receive (contains) the specified element. After updating this subset, assuming at least one subset is critical, some global maintenance is performed to insure preservation of the invariant condition. A set of size $\lfloor n/k(n)\rfloor$ (or smaller, if one exists) is merged with one of its neighbouring subsets and the resulting set is inserted back into the priority queue. Next, a set of size $2\lfloor n/k(n)\rfloor + 2$ (or larger, if one exists) is split into two (almost) equal sized subsets, both of which are inserted back into the queue.

Since each update involves at most one dictionary reference and a constant number of priority queue operations both on sets of size $O(n)$, the result follows immediately. □

5.2 A More General Dynamization Result

If we let lemma 5.1 replace lemma 2.4 in the argument of Section 2.2, it follows with some (but not a great deal) more work (see [6] for details) that our technique for trading deletion cost for storage will apply to data structures that can be merged (exploiting "separability" if necessary) faster than they can be totally reconstructed. As we have already indicated this permits a number of new applications of our basic technique. Some of these applications are described in the next section. For further details and applications the reader is referred, once again, to [6].

5.3 Two Applications

We have already mentioned the result of Preparata and Hong [12] that convex hulls of planarly separable point sets in 3-space can be merged (into the convex hull of their union) in linear time. (Note, it is not clear whether separability is necessary for this result). This leads directly to the following:

Application 5.2 The convex hull of a set of n points in 3-space can be maintained dynamically at a cost of $O(n)$ steps per insertion or deletion.

It should be noted that the entire convex hull (as opposed to pieces or "profiles") is maintained by this scheme, making possible not only fast dynamic hull inclusion queries, but also fast dynamic maintenance of structures directly related to the convex hull.

A particularly interesting application of 3-d convex hulls is their use in the representation of the common intersection of 3-d half spaces, using a transform of Brown [3]. It turns out that by careful exploitation of the properties of this transformation (and a modification of the 3-d convex hull algorithm of [12]) we are able to claim the following:

Application 5.3 The common intersection of a set of n 3-d half spaces can be maintained dynamically at a cost of O(n) steps per insertion or deletion (of a half space).

6. CONCLUSIONS

We have presented a technique for improving the worst case cost of dynamic maintenance for a broad class of data structures. Specifically, we have shown how the cost of performing deletions on "linearly mergeable" structures can be reduced at the expense of extra storage. Our technique leads to improved dynamic algorithms for the maintenance of nearest (and furthest) point Voronoi diagrams, nearest (and furthest) neighbour search structures, 3-d convex hulls, and the common intersection of 3-d half spaces.

REFERENCES

[1] Aho, A.V., Hopcroft, J.E. and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley (1974).

[2] Bentley, J.L., "Decomposable Searching Problems", Info. Proc. Lett. 8, 5 (1979), 244-251.

[3] Brown, K.Q., "Geometric Transforms for Fast Geometric Algorithms", Tech. Rep. CMU-CS-80-101, Carnegie-Mellon University (1979).

[4] Cheriton, D. and Tarjan, R.E., "Finding Minimum Spanning Trees", SIAM J. Comput. 5,4 (1976), 724-742.

[5] Edelsbrunner, H., "Optimizing the Dynamization of Decomposable Searching Problems", Report 35, Inst. fur Informationsverarbeitung, TU Graz, Austria (1979).

[6] Gowda, I.G., "Dynamic Problems in Computational Geometry", M.Sc. thesis, University of British Columbia (1980).

[7] Kirkpatrick, D.G., "Efficient Computation of Continuous Skeletons", Proc. 20th Annual IEEE Symp. on Foundations of Computer Science, (1979) 18-27.

[8] Kirkpatrick, D.G., "Optimal Search in Planar Subdivisions", preprint, Department of Computer Science, UBC, Vancouver, Canada (1979).

[9] Lipton, R.J. and Tarjan, R.E., "Applications of a Planar Separator Theorem", Proc. 18th Annual IEEE Symp. on Foundations of Computer Science (1977), 162-170.

[10] Maurer, H.A. and Ottman, Th., "Dynamic Solutions of Decomposable Searching Problems", Report 33, Institut fur Informationsverarbeitung, TU Graz, Austria (1979).

[11] Overmars, M.H. and van Leeuwen, J., "Two General Methods for Dynamizing Decomposable Searching Problems", Tech. Rep. RUU-CS-79-10, Department of Computer Science, University of Utrecht (1979).

[12] Preparata, F.P. and Hong, S.J., "Convex Hulls of Finite Sets of Points in Two and Three Dimensions", Comm. ACM 20, 2 (1977), 87-93.

[13] Saxe, J.B. and Bentley, J.L., "Transforming Static Data Structures to Dynamic Structures", Proc. 20th Annual IEEE Symp. on Foundations of Computer Science (1979), 148-168.

[14] Shamos, M.I., "Computational Geometry", Doctoral dissertation, Yale University (1978).

[15] Shamos, M.I. and Hoey, D., "Closest-Point Problems", Proc. 16th Annual IEEE Symp. on Foundations of Computer Science (1975), 151-162.

[16] van Leeuwen, J. and Wood, D., "Dynamization of Decomposable Searching Problems", Info. Proc. Lett. 10,2 (1980), 51-56.

Algorithms for Testing the Inclusion of Points in Polygons[1]

Jon Louis Bentley
Wendy Carruthers[2]
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract -- Determining whether a given point lies inside or outside a simple polygon is an important problem in many applications, including computer vision systems and computer-assisted political redistricting systems. In this paper we give algorithms for inclusion problems that are efficient for polygons that are "close to convex" in a certain precise sense. An empirical study of polygons that arise in several applications shows that typical polygons are indeed "close to convex", and a program implementing the algorithm shows that is extremely efficient on point sets of practical sizes.

1. Introduction

The problem of determining whether a point is inside or outside a simple polygon arises in many applications. In a computer-assisted legislative redistricting system, for instance, the point represents a precinct centroid and the polygon represents a proposed legislative district, and we want to determine whether this precinct is within the proposed district (see Heslop [1980] for details of this application). In a computer vision system, the polygons represent identified regions of the picture, and the user can ask in what region the given query point lies. There are three distinct computational problems motivated by such applications:

- **The Single-Shot Problem:** Given a polygon of N vertices and a single point, determine whether the point is inside or outside the polygon.
- **The Searching Problem:** Organize a polygon of N vertices such that subsequent queries asking whether a given point is inside or outside the polygon can be answered quickly.
- **The Batched Problem:** Given a set of M points and a polygon of N vertices, report for each point whether it is inside or outside the polygon.

In this paper we shall study one general computational technique that can be used to solve all the above problems, along with several others.

Much previous work has been done on the problem of determining point inclusion in simple polygons. Shamos [1978] describes a number of approaches to the problem, including single-shot determination, and simple and efficient searching algorithms for convex and star-shaped polygons. Lipton and Tarjan [1977] gave an asymptotically efficient algorithm for the searching problem, but it is too complex to code for most applications; Preparata [1979] later gave an

---

[1] This research was supported in part by the Office of Naval Research under contract N00014-76-C-0370 and in part by the National Science Foundation under Grant MCS-7825824.

[2] This research was performed while this author was visiting Carnegie-Mellon University. Her current address is Department of Mathematics, Bucknell University, Lewisburg, Pennsylvania 17837.

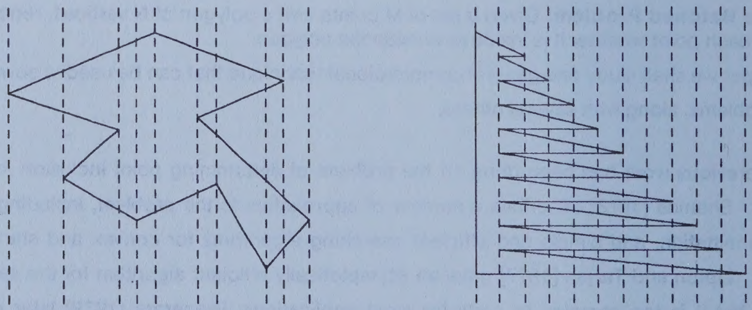
asymptotically less efficient but more practical algorithm for the same problem.

In this paper we will investigate a set of algorithms that are very efficient for a broad class of polygons (including almost all polygons that arise in practice). In Section 2 we will study particular algorithms for solving the searching problem. The techniques we study in that section are extended in Section 3 to solve the batched problem and several other polygon problems. In Section 4 we discuss our experience in using the algorithms on "real" polygons, and conclusions are then offered in Section 5.

2. The Searching Problem

Experience has shown that polygon problems are much easier when they are restricted to convex polygons; recall that a polygon is convex if any straight line intersects it at most once. Although many polygons that arise in practice are not convex, they are close to convex in the following sense: any vertical line intersects the polygon only relatively few times. In this section we will study an algorithm for the searching problem that is particularly efficient for polygons that are close to convex in this sense. This algorithm was originally described by Shamos [1978, Section 4.4], but he analyzed its performance only in a worst-case sense, and showed that it can perform quite poorly.

Recall that in the searching problem we must organize the N vertices of the simple polygon P such that we can quickly determine whether a new point lies within P. The fundamental idea of Shamos's data structure is easy to understand intuitively, and is illustrated on two polygons in Figure 1. To organize the polygon, we simply "draw a vertical line" through each of its vertices. This divides the plane into at most vertical N + 1 slabs. When we are given a new query point, we first determine in which vertical slab it lies; this can be quickly accomplished by a binary search by x-value. We must now determine whether it is inside or outside the polygon, but this is very easy by the regular structure of the polygon within each slab.



a.) A typical polygon.

b.) A worst-case polygon.

Figure 1. A data structure illustrated on two polygons.

We will now investigate the details of the method just sketched. The input to the procedure is a simple polygon, given as a vector containing the x and y coordinates of the vertices in counterclockwise order. The first phase of the preprocessing step sorts the x-values of the vertices; the at most N distinct x values partition the plane into at most N+1 slabs. We will represent these slabs by a vector, sorted by x-values. Each element of the vector contains both the x-value of its upper limit and a pointer to a structure representing the edges that intersect that slab, *sorted in increasing order by y-value.* The simplicity of the polygon ensures that an "aboveness/belowness" relation exists between all edges active in a given slab; we can compare the y-values of the edges at the slab's middle x-value, for instance.

Given the above data structure, it is easy to test whether a given point A lies inside or outside polygon P. If A's x-value is not within the polygon's range of x-values, we report that it is outside; otherwise, we use a binary search to determine in which slab A lies. We can then do a binary search among the sorted edges to find in which region of the slab A lies, and then check a single bit to see whether it is inside or outside the polygon. (It is inside if it is above an odd number of the line segments.) Note that this method also makes it easy for us to report the fact that A happens to lie on an edge of the polygon. Because this searching procedure consists only of two binary searches, each on a set of at most N+1 elements, it requires at most $2\lceil \lg N+1 \rceil$ comparisons.

The preprocessing algorithm that builds this data structure is quite simple. Its first step is to sort the vertices of the polygon in increasing order by x-coordinate. It then uses that list to "scan through" the polygon, left-to-right, always keeping track of the edges intersecting the current "scan line". At every vertex the algorithm does two things: write the current "slice" of the polygon into the data structure, and modify the current view of the slice. There are three cases to consider when meeting a new vertex: it can be the start of a new portion of the polygon, it can be the end of a current portion, or it can be a change of direction of an existing edge. All of these modifications can be dealt with easily be storing the edges intersecting the current scan line in a sorted data structure. The details of this method have been discussed by Shamos [1978, Section 4.4], who also gave a psuedo-Algol program to implement this method.

Shamos has given a worst-case analysis of the three important aspects of the above structure: preprocessing time, query time, and storage. When the structure represents a polygon of N vertices, we denote the three above quantities by the functions P(N), Q(N), and S(N), respectively. We already mentioned that because the query performs only two binary searches, its worst-case performance is proportional to lg N. To analyze the worst-case preprocessing and storage requirements, we need only consider the performance of the algorithm on the worst-case polygon shown in Figure 1.b. Since there are an average of N/4 portions of the polygon active in each slab, both the storage and preprocessing must grow at least as quickly as $\theta(N^2)$. Shamos has shown that these are also worst-case upper bounds. These facts together establish the performance of the algorithm as

$P(N) = \theta(N^2)$
$Q(N) = \theta(\lg N)$, and
$S(N) = \theta(N^2)$.

The above analysis, though accurate in the worst case, is incredibly pessimistic: it assumes that the user will want to process pathological polygons. In most applications, though, we know that the polygons we see are not like that in Figure 1.b, and we would like some way to quantify that knowledge. The key observation to be made is that the polygon of Figure 1.b is "bad" because one vertical line intersects it many times, while polygons that arise in most applications are "good" because one vertical line intersects them only a few times. The following definitions allow us to capture that property more precisely.

Definitions: Column Convexity

The *column convexity* of polygon P at x-value R is the number of connected regions contained in the intersection of P and the vertical line X = R. (Thus the column convexity of a convex polygon is one at all x-values in its range.) The maximum column convexity of polygon P, denoted by $k_m(P)$ or k_m when P is obvious by context, is the maximum column convexity of all x-values. The *unweighted mean column convexity*, k_u, is taken to be the average column convexity over all (up to) N-1 slabs in the polygon. The *weighted mean column convexity*, k_w, is the average column convexity over all slabs in which the column convexity of each slab is weighted by its x-width.

These definitions will allow us to analyze the performance of the data structure in terms of the represented polygon's true column convexity, rather than the worst of all possible column convexities. For instance, the storage used by Shamos's data structure is clearly $O(Nk_m)$, because the number of edges stored in any slab is bounded above by twice k_m, the maximum column convexity. Similarly, the preprocessing time can be shown to have worst-case performance $O(N \lg N + Nk_m)$, where the first term is due to sorting the N vertices, and the second term accounts for scanning through the set and actually storing the data structure. In Section 4 we will see that in many applications, polygons have a maximum column convexity of, say, five. In such applications, the data structure of this section can be quite practical.

We can use the definitions of average column convexity to sharpen the above results. Note that the number of edges stored in the structure is exactly $2Nk_u$; hence the unweighted average column convexity is a useful predictor of storage use. The weighted average column convexity tells us how many edges, on the average, the vertical binary search must investigate if we assume that query points have their x-values uniformly distributed along the x-range of the polygon; we will return to this issue in Section 4.

3. Other Problems

In the previous section we saw an algorithm for the searching problem that was based on two fundamental operations: dividing a polygon into slabs and scanning through the polygon. In this section we will see how other problems can be solved by applying the same two techniques.

The first problem that we will consider is that of batched point inclusion, in which we are given an N-vertex polygon and we must report whether each of M query points is inside or outside the polygon. While the searching problem had to preprocess the polygon to allow arbitrary points to be tested, in this problem we have an *a priori* knowledge of all the points. We will solve this problem by an algorithm based on slabs and scanning, which is roughly described as follows. We will scan through the polygon, left-to-right, always keeping track of the part of the polygon intersecting the current vertical scan line. Whenever we encounter a point, we solve the one-dimensional problem of determining whether its y-value lies in an active region of the current scan line.

We will now describe the algorithm more precisely. Its inputs are the N vertices of the polygon and the M query points. The first operation of the algorithm creates a data structure representing all $M+N$ points sorted by x-value, with each point marked as a vertex or query point. We then use that structure to implement the left-to-right scan. As we encounter a query point in the scan, we test whether it is inside or outside the current active region of the polygon (we will refine this test later). There are three cases to consider as we encounter a polygon point:

1. It is the start of a new part of the polygon (that is, a leftmost vertex), in which case we add a new entry to the active region.
2. It is the end of an old part of the polygon (that is, a rightmost vertex), in which case we delete an existing portion of the active region.
3. It is a change of direction of the polygon, in which case we modify the active region.

Notice that the scan of this algorithm is exactly the same as the preprocessing phase of the previous section, and the active region corresponds precisely to a slab in the previous data structure.

We have yet to specify an implementation of the active region of the polygon (that is, a representation of the edges intersecting the current scan line). We must keep a set of nonintersecting line segments sorted by y-order, and two structures come to mind for this task: a sorted linked list and a balanced binary tree. Although the binary tree is superior for establishing an efficient asymptotic performance, the linked list is usually the method of choice for applications.

We turn now to an analysis of the algorithm. Sorting the points in a straightforward manner can be done in time proportional to $N \lg N + M \lg M$ (although Hong Jia-Wei recently proved a conjecture of the authors that the time required to sort the polygon's vertices can be reduced to $\theta(N \lg k_m)$). The scan will then take time proportional to $M+N$ plus the time required to manipulate the active region, which contains a maximum of $2k_m$ segments. If a balanced tree is used, then each operation will have time bounded above by $\theta(\lg k_m)$; if a linked list is employed, then each operation requires time proportional to k_m. Thus the worst case time of the algorithm is

$$O(N \lg N + M \lg M + (N+M) \lg k_m) = O((N+M) \lg (N+M))$$

if the active region is implemented as a balanced binary tree.

The average performance of the linked list implementation of the average region is usually very good. The worst-case scan time of this method is $\theta(Nk_m)$, but if the points are uniformly distributed in x-value across the x-region of the polygon, then the average scan time is exactly

$\theta(Nk_w)$. (Note that the weighted mean column convexity gives the expected number of regions of the polygon intersecting a scan line whose x-value is chosen uniformly over the polygon's x-extremes.)

The above algorithm can be extended in several ways. For instance, one could implement it to answer a batched query as it organizes the polygon and simultaneously preprocess it so it can quickly answer a later search. The basic algorithm can easily be augmented to check the input polygon for self intersection during the scan (if two edges intersect, they do so at a particular scan line); this feature actually found several bugs in the input polygons of the next section. One can also modify the searching algorithm, for example, to store several polygons at once and tell which of those contain a particular query point.

In the next section we will see that most polygons that arise in many applications have very low column convexity; this implies that most of the work in the above algorithms is devoted to processing the x-values of the polygons and the points. If those x-values are known to be smoothly distributed along their range, then the techniques of Weide [1978] can be used to reduce the time required to deal with them. Specifically, Weide shows that N points can be sorted in linear expected time and that a "slab lookup" can be accomplished in constant expected time for a broad class of distributions of x-values. Applying these techniques to the above problems yields optimal expected-time algorithms.

4. Data on Implementations

So far in this paper we have viewed the algorithms only as theoretical tools to establish order-of-magnitude bounds on the computational complexity of geometric problems. In this section we will see that they can in fact be used to solve practical problems very efficiently. There are two distinct aspects in this verification of their utility: we must first show that polygons that arise in practice do in fact display low column convexity, and secondly we must show that the algorithms can be coded to run quickly on problems of practical size.

We will now investigate the column convexity of several different classes of polygons that arise in various applications. As the first class, we will study the small but important set of polygons that represent different characters in a computerized (raster) character font. Note that all edges of the polygon corresponding to a raster character are either vertical or horizontal; while some of the polygons are not simple (consider "i" and "o", which are disconnected and have genus one, respectively), we will nonetheless consider the polygons as wholes. We will examine the News Gothic 12 font used by the Carnegie-Mellon University Computer Science Department Xerox Graphics Printer. Among the fifty-two upper and lower case letters, the maximum column convexity is four, which is displayed by lower case "g" (note that this report is not printed with a News Gothic Font!). Several letters have maximum column convexity of three (e, s, z, B, E, G, Q, S and Z), while many have a maximum column convexity of unity (I, v, w, L, M, N, T, U, V, W, and Y). Of the special characters, several have maximum column convexity of four (8, {, }, @, and

&), and many display a maximum of three. This font seems to be typical of many fonts: although the number of edges in the polygons may be in the hundreds, the maximum column convexity is only four. Thus the algorithms in this paper would be quite efficient in applications dealing with character generation and recognition.

The next problem domain that we investigated consisted of polygons that arose in an AI vision system. Specifically, we studied ten polygons that were digitized representations of regions in aerial photographs of the Washington, D.C. area. Table 1 shows the number of vertices in each polygon and the mean, weighted mean, and maximum column convexity of the polygons. It shows that all of the values are very small when compared to the maximum possible column convexity of N-vertex polygons.

N	Weighted Mean	Unweighted Mean	Max	N	Weighted Mean	Unweighted Mean	Max
369	1.75	1.85	4	1229	1.9	2.35	5
377	2.90	2.90	6	1682	3.6	4.40	8
403	3.03	3.25	7	1735	2.35	2.85	6
405	1.85	1.90	5	2180	2.1	2.6	9
447	4.45	4.55	8	3576	6.55	7.45	15

Table 1. Column convexities of ten vision polygons.

We also briefly studied the column convexity of polygons in geopolitical applications. For instance, the appendix of Heslop [1980] is entitled "Examples of Legislative Gerrymanders" and contains five polygons that represent particularly perverse legislative districts in the 1972 reapportionment of California (Assembly Bill Number 12). The maximum column convexity of any of the districts in six; it occurs in Assembly District 69, which contains portions of thirteen cities and two counties, but properly contains no cities! (It was created to elect a Democratic representative in predominantly Republican Orange county.) The remaining four districts, although all quite perverse, all had maximum column convexity less than six (five, five, four and three). We also examined several other kinds of geopolitical polygons (such as states of the union and zip code regions), and all of them displayed very small column convexity.

At this point we would like to make a very specific claim: we studied a set of polygons that users indicated were typical, and all the polygons had very low column convexity; specifically, they all had maximum column convexity less than twenty, and usually less than five. We therefore feel safe to assume that in the three applications of character handling, vision systems, and geopolitical databases, most polygons have very low column convexity. Note that we are not claiming that all polygons have this property; but the ones that arose in the applications we were studying certainly did.

To test the practicality of the scanning methods of this paper we coded both the batched and searching algorithms in the C programming language on a VAX-11/780, and then ran the polygons on the ten vision polygons discussed above. We will now briefly describe the timing values of the scanning algorithm for the batched problem; the performance of the searching algorithm was quite comparable. To test the inclusion of N points in an N-vertex polygon, the running time of the scanning algorithm was observed to be (.483 N lg N - 2.8N) milliseconds for the sorting phase (a Quicksort), and Nk_w milliseconds for the actual scanning (where k_w denotes the average column convexity of the polygon). The current scan line was implemented as a sorted linked list. As a "straw man", we also implemented the angle method for determining point inclusion (see Shamos [1978, Section 4.4]); its running time to test the inclusion of a single point in an N-vertex polygon was observed to be 2.7N milliseconds. To gather data on an N-point vision polygon we generated N points at random uniform over the bounding box of the polygon, and then tested that batch for inclusion by the scanning algorithm. To determine the run time of the angle method on a particular polygon we tested 100 points, and used the average time as the estimate of running time; testing the entire batch by the angle method would have required prohibitive amounts of time. This process was applied to all ten of the vision polygons, and the average scanning times were fitted to the analytic model of Section 3 to yield the predictions reported above. To give a better feeling for the two methods, Table 2 gives details of the experiments on the smallest and the two largest of the polygons.

Polygon	Smallest	Second	Largest
N	369	2180	3576
Weighted Mean Convexity	1.75	2.1	6.55
Angle Method Times			
One Point (Seconds)			
Predicted	.99	5.89	9.65
Observed	.8	6.2	9.67
N Points Predicted	5 min.	3.75 hrs.	9.6 hrs.
Scanning Method Time			
Sort Time (Seconds)			
Predicted	.49	5.8	10.5
Observed	.45	5.5	11.75
Scan Time (Seconds)			
Predicted	.62	4.5	17.0
Observed	.7	5.0	20.2
Total Observed Time	1.15 sec.	10.5 sec.	31.95 sec.

Table 2. Test N points in an N-vertex polygon.

Table 2 tells two simple stories. First, the predictors are usually within ten percent of the observed values (this was also true for the remaining seven polygons). And secondly, the scanning algorithm is *much* faster than its naive competitor, the angle algorithm. The underlined values of predicted time for the angle method and observed time for the scanning algorithm offer a striking comparison: for the largest polygon, the scanning algorithm solves the problem in only 32 seconds, while the angle algorithm would require over 9 hours!

5. Conclusions

We feel that there are three primary contributions of this paper. The first contribution is the algorithms, which establish new bounds on the expected complexity of solving various polygon problems. The second contribution is the program corresponding to the algorithms: the data in Section 4 shows that it is an eminently practical tool. The third contribution is the type of analysis we have done: we have identified a particular property of polygons (low column convexity) that (1) occurs frequently in real problems, (2) is easy to test, and (3) can be efficiently exploited algorithmically. We feel that the identification of such properties will prove to be an important tool in bridging the existing gap between algorithm design and applications programming.

Acknowledgments

Helpful conversations with David McKeown are gratefully acknowledged.

References

Heslop, A. [1980]. "Redistricting: The key to politics in the 1980's," Report, Rose Institute of State and Local Government, Claremont Men's College, Claremont, California.

Lipton, R. J. and R. E. Tarjan [1977]. "Applications of a planar separator theorem," *Proceedings of the Eighteenth Symposium on the Foundations of Computer Science*, pp. 162-170.

Nievergelt, J. and F. P. Preparata [1980]. "Plane-sweep algorithms for intersecting geometric figures," University of Illinois Coordinate Science Laboratory Report R-863, October 1979.

Preparata, F. P. [1979]. "A new approach to planar point location," University of Illinois Technical Report, August 1979.

Shamos, M. I. [1978]. Computational geometry, unpublished Ph. D. thesis, Yale University, May 1978.

Weide, B. W. [1978]. Statistical methods in algorithm design and analysis, unpublished Ph. D. thesis, Carnegie-Mellon University, August 1978.

DECOMPOSING A SIMPLE POLYGON WITH THE RELATIVE NEIGHBORHOOD GRAPH

GODFRIED T. TOUSSAINT
School of Computer Science, McGill University, Montreal, Quebec H3A 2K6

ABSTRACT

The decomposition of a polygon into simpler components plays an important role in syntactic pattern recognition and image processing. In this paper a new decomposition is proposed and termed the *relative neighbor decomposition* (RND). The lune of two vertices p_i, p_j of a polygon P, denoted by LUNE (p_i,p_j) is defined as the intersection of two circles with radius equal to $d(p_i,p_j)$ centered at p_i and p_j, where d denotes euclidean distance. The RND consists of the polygon P together with a subset of its diagonals. Two vertices p_i,p_j are joined by a diagonal iff (a) p_i and p_j are *visible* to each other and (b) no other *visible* vertex of p_i or p_j lies inside LUNE (p_i,p_j). It is shown that the RND is planar, i.e., no two diagonals intersect. The straightforward approach to computing the RND of P leads to an $O(n^4)$ algorithm, where n is the number of vertices of the polygon. However, one can use the *viewability graph* VG(P) to obtain an algorithm that runs in O(ne) time, where e is the number of edges in VG(P) and where $O(n) \leq e \leq O(n^2)$.

1. INTRODUCTION

The decomposition of a simple planar polygon into simpler components plays an important role in syntactic pattern recognition. Some examples of possible decompositions are: decompositions into convex polygons [1], [2], [3], decompositions into spiral polygons [4], decompositions into monotone polygons [5], and decompositions into star polygons [6]. A survey of these methods and many other additional references are contained in [7].

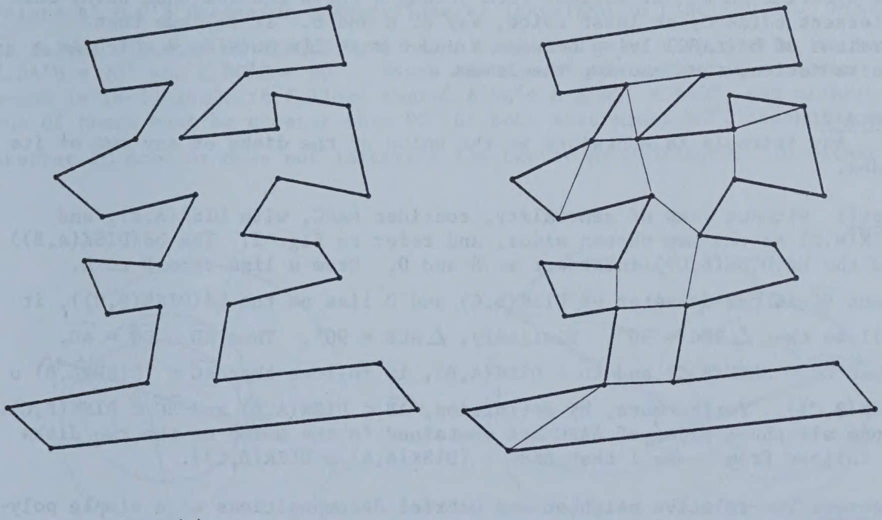
In this paper we propose a new decomposition, termed the *relative neighbor decomposition* (RND), that uses the idea of the *relative neighborhood graph* (RNG) of a planar set of points to determine which diagonals of a polygon P should be added to P to form the RND(P). We then exhibit a proof that the RND is planar, i.e., that the simpler components of P do not overlap. Next, we show how the *viewability graph* of P can be used to obtain an algorithm for computing the RND of an n-vertex polygon in $O(n^3)$ time, and finally we close with some open problems.

2. THE RND PROBLEM

Consider a set of n distinct points on the plane: $\mathcal{P} = \{p_1,p_2,\ldots,p_n\}$ given by their cartesian coordinates. The lune of two points p_i,p_j, denoted by $\mathrm{LUNE}(p_i,p_j)$, is defined as the intersection of two circles with radius equal to $d(p_i,p_j)$ centered at p_i and p_j, where d denotes euclidean distance. The RNG is a planar graph on $\mathcal{P}$ with the property that p_i,p_j forms an edge of the RNG iff no other point lies in the interior of $\mathrm{LUNE}(p_i,p_j)$ [8]. The straightforward approach to computing the RNG leads to an $O(n^3)$ algorithm but in [8] an $O(n^2)$ algorithm is presented which uses the Voronoi diagram. An $O(n \log n)$ algorithm was recently exhibited in [9] but it is easy to show that it does not always work [10]. Even more recently Supowit [11], in a detailed study of the RNG, presented an $O(n \log n)$ algorithm for the RNG of a planar set and an $O(n)$ algorithm for the RNG of a convex polygon.

To obtain the *relative neighbor decomposition* (RND) of a simple polygon P we extend the definition of the RNG as follows: two vertices of P, say p_i, p_j are joined by a diagonal iff 1) p_i and p_j are *visible* and 2) no other *visible* vertex of p_i or p_j lies inside $\text{LUNE}(p_i,p_j)$. Two vertices p_i and p_j of P are *visible* iff the edge p_i,p_j lies in P. Figure 1 shows a simple polygon and its RND.

A graph which is closely related to the RNG is the Gabriel graph. The disk of p_i and p_j, denoted by $\text{DISK}(p_i,p_j)$ is defined as follows: $\text{DISK}(p_i,p_j) \equiv \{x : [d^2(x,p_i) + d^2(x,p_j)]^{1/2} \leq d(p_i,p_j)\}$. Clearly, p_i and p_j lie on a circle and determine its diameter. This circle is known also as the "circle of influence" and is an alternate definition of relative neighborhood to that specified by the "lune of influence" used in the RNG. Since the disk is a subset of the lune it follows that the RNG is a subgraph of the Gabriel graph (GG). Matula and Sokal [12] derive many interesting properties of Gabriel graphs and show how it can be computed in $O(n \log n)$ time using the Voronoi diagram. Clearly, this idea also translates to the Gabriel decomposition of a polygon (GGD) by substituting the word LUNE by DISK and joining two *visible* vertices p_i,p_j by an edge iff no other *visible* vertex of p_i or p_j lies in $\text{DISK}(p_i,p_j)$.



(a) (b)

Fig. 1. (a) Simple polygon obtained by image processing technique applied to a Chinese character [4]. (b) Relative neighbor decomposition of polygon in (a) ; a post-processed version appears in Fig. 7.

As expected, this decomposition yields many more components than the RND and, while it is interesting in its own right, it does not appear to be as useful as the RND for pattern recognition applications, where we want the decomposition to contain a relatively small number of components.

It is known that the RNG and the GG of a set of points are planar graphs [8], [12]. However, it does not follow from this that the RND and GGD are planar, since the latter graphs may contain edges with non-empty lunes and disks, as long as any vertices contained in, say, LUNE(p_i,p_j) are not visible from either p_i or p_j. In other words the RND of P may contain edges not contained in the RNG of the vertices of P. Thus the first most important question that arises concerning these decompositions is their planarity. It follows trivially that if the GGD is planar then so is the RND. In the next section we show that the GGD is planar.

3. PROOF OF PLANARITY

First we introduce some notation. Two points on the plane A and B will be denoted by (A,B). An edge or line segment with endpoints A and B will be denoted by $\overline{AB}$. A triangle with vertices A,B, and C will be denoted by ΔABC, whereas $\angle ABC$ will denote the angle at vertex B formed by vertices A and C. Let bd(DISK(A,B)) denote the boundary of DISK(A,B). We will also use CIRCLE(A,B) to denote bd(DISK(A,B)). The interior of DISK(A,B) will be denoted by int(DISK(A,B)) and thus does not include the circle itself.

Lemma 1:

If the three sides of a triangle ABC lie in a simply connected region R, then int(ΔABC) also lies in R.

Proof: Let $\overline{AB}$, $\overline{BC}$, $\overline{CA}$ all lie in R and let y be a point interior to ΔABC and exterior to R. It follows from Jordan's curve theorem that bd(R) must intersect bd(ΔABC) at least twice, say at a and b. It follows that portions of bd(ΔABC) lying between a and b must lie outside R which is a contradiction, thus proving the lemma.

Lemma 2:

Any triangle is contained in the union of the disks of any two of its sides.

Proof: Without loss of generality, consider ΔABC, with DISK(A,B), and DISK(B,C) as the two chosen sides, and refer to Fig. 2. The bd(DISK(A,B)) and the bd(DISK(B,C)) intersect at B and D. Draw a line from B to D. Since $\overline{BC}$ is the diameter of DISK(B,C) and D lies on the bd(DISK(B,C)), it follows that $\angle BDC = 90°$. Similarly, $\angle ADB = 90°$. Thus $\overline{AD} \cup \overline{DC} = \overline{AC}$. Since $\overline{DC} \subset$ DISK(B,C) and $\overline{AD} \subset$ DISK(A,B), it follows that $\overline{AC} \subset$ {DISK(A,B) $\cup$ DISK(B,C)}. Furthermore, by definition, $\overline{AB} \subset$ DISK(A,B) and $\overline{BC} \subset$ DISK(B,C). Since all three sides of ΔABC are contained in the union of the two disks it follows from Lemma 1 that $\Delta ABC \subset$ {DISK(A,B) $\cup$ DISK(B,C)}.

Theorem: The relative neighbor and Gabriel decompositions of a simple polygon P are planar; i.e., no two diagonals of the RND or the GGD intersect each other or the edges of P.

Proof: (by contradiction for the GGD case)

We shall begin by assuming that two given diagonals of the GGD, $\overline{AC}$ and

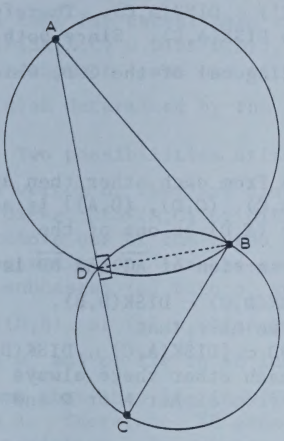


Fig. 2

$\overline{BD}$, intersect and then proceed to show that this cannot be true. Two cases arise: either all vertices A,B,C,D are visible from each other (case A), or at least one pair of vertices of the set $\{(A,B), (B,C), (C,D), (D,A)\}$ is such that its two vertices are not visible to each other (Case B). Note that since $\overline{AC}$ and $\overline{BD}$ are diagonals of the GGD, A and C are visible to each other and so are B and D. It follows that no diagonals intersect polygon edges.

Case A:

Consider diagonal $\overline{DB}$ and DISK(D,B). The position of the remaining two vertices A and C, with respect to DISK(D,B), determines four possible situations: both A and C lie in DISK (D,B) , only A does, only D does, and finally neither of the two lie in DISK(D,B). In the first three cases it is immediate that at least one visible vertex of either D or B lies in DISK(D,B) and hence $\overline{DB}$ cannot be a diagonal of the GGD, which is a contradiction. For the case when both A and C lie outside of DISK(D,B) refer to Fig. 3. Since $\overline{AC}$ intersects $\overline{DB}$ and $\overline{DB} \subset$ DISK(D,B), it follows that $\overline{AC}$ intersects DISK(D,B). Let A' and C' denote the points where $\overline{AC}$ intersects DISK(D,B) and construct DISK(A',C'). Points A',D,C'B determine a quadrilateral inscribed in CIRCLE(D,B). Since $\overline{DB}$ is the diameter of CIRCLE(D,B) and A' and C' lie on this circle, we have $\angle DA'B = 90°$ and $\angle DC'B = 90°$. Since the sum of the angles of a convex n-gon is $(n-2)\times 180°$, it follows that $\angle A'DC' + \angle A'BC' = 180°$, and either one of these must be greater than 90° or both must equal 90°, depending on whether $\overline{AC}$ does or does not intersect the center of DISK(D,B). In either

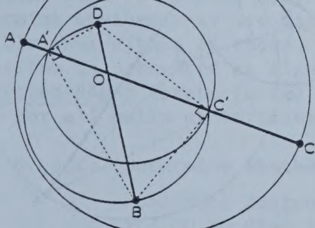


Fig. 3

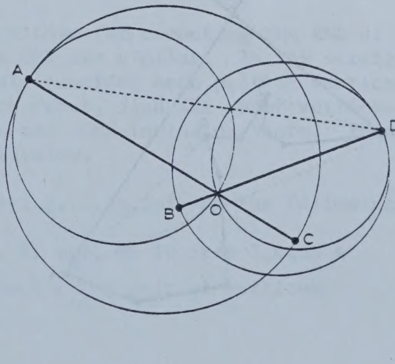


Fig. 4

case, since $\overline{A'C'}$ is the diameter of DISK(A',C'), at least one of the two vertices D,B (the one whose angle is greater than 90°) must lie in DISK(A',C'). Since $\overline{A'C'} \subset \overline{AC}$, we have DISK(A',C') ⊂ DISK(A,C). Therefore at least one of the two vertices D,B must lie in DISK(A,C). Since both D and B are visible from A and C, $\overline{AC}$ cannot be a diagonal of the GGD, which is a contradiction.

Case B:

If not all the vertices A,B,C,D are visible from each other then at least one pair of vertices of the set {(A,B), (B,C), (C,D), (D,A)} is an invisible pair. Without loss of generality, let (A,D) be one of the invisible pairs, and refer to Fig. 4. The intersection of $\overline{AC}$ and $\overline{BD}$ is denoted by O. Since $\overline{DO} \subset \overline{DB}$ it follows that DISK(D,O) ⊂ DISK(D,B). Similarly, DISK(A,O) ⊂ DISK(A,C). From lemma 2 we have that Δ AOD ⊂ [DISK(A,O) ∪ DISK(O,D)]. Therefore, ΔAOD ⊂ [DISK(A,C) ∪ DISK(B,D)]. We will now show if A and D are not visible to each other there always exists at least one vertex of P that is visible from either A or D and that lies in DISK(A,C) ∪ DISK(B,D).

Since (A,C) and (B,D) are visible pairs, because they are diagonals of the GGD of P, neither of the two polygonal chains (A,...,D) of P can intersect $\overline{AO}$ or $\overline{OD}$. Now, if A and D are not visible, it follows that at least one of the chains (A,...,D) must be either completely contained in Δ AOD, or it must intersect $\overline{AD}$ at least once. In either case at least one vertex of one chain (A,...,D) other than A or D must lie in int(Δ AOD). With $\overline{DA}$ as the x-axis and D the origin consider the *upper convex hull* of the set of vertices consisting of D,A, and the vertices of the chains (A,...,D) that lie in int (Δ AOD), Fig. 5. On this convex polygonal chain let p_i be the vertex adjacent to A and let p_j be the vertex adjacent to D. Note that it is possible that $p_i = p_j$. Since $(\overline{DO} \cup \overline{OA}) \subset P$, the polygon $(D,O,A,p_i,\ldots,p_j,D) \subset P$, and no part of the remaining polygonal chain can

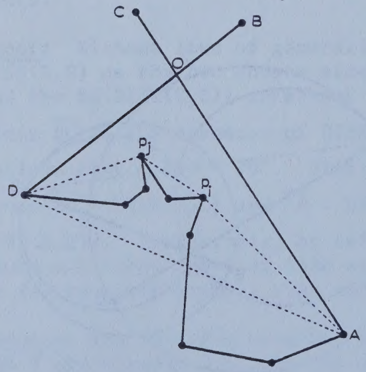


Fig. 5

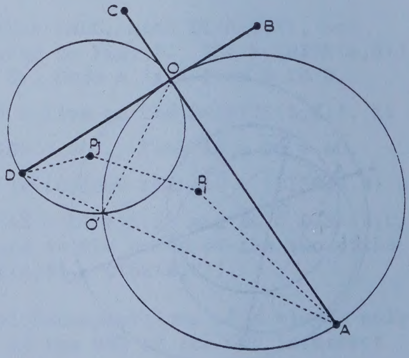


Fig. 6

lie in $(D,O,A,p_i,\ldots,p_j,D)$. Therefore $\overline{Dp_j} \subset P$ and $\overline{Ap_i} \subset P$ and therefore p_i is a vertex visible from A whereas p_j is a vertex visible from D. We have thus far established that there must exist vertices in ΔDOA, and hence in [DISK(A,C) $\cup$ DISK(B,D)], that are visible from A or D. It remains to be shown that at least one of the two vertices p_i,p_j is always contained in the disk determined by the vertex from which it is visible.

Two possibilities arize for p_i and p_j. Either they are one and the same vertex or they are distinct. In the first case it must lie in one of the disks, DISK(A,C) or DISK(B,D) and it is visible from both A and D. Therefore one of the edges $\overline{AC}$ or $\overline{BD}$ cannot be a diagonal of the GGD of P, a contradiction. In the second case, where p_i and p_j are distinct, we have two subcases: (a) both p_i and p_j lie in at least one of the disks DISK(A,C), DISK(D,B), or (b) p_i lies in one disk and p_j in the other. In subcase (a) both p_i and p_j lie in at least one of the disks; without loss of generality, assume that they lie in DISK(A,C). One of the two vertices must be visible from A. Therefore, $\overline{AC}$ cannot be a diagonal of the GGD(P) and we have a contradiction. In subcase (b) we must show that we cannot have a situation where $p_i \subset$ DISK(D,B) and not DISK(A,C), and correspondingly for p_j. Let the intersection of CIRCLE(O,D) and CIRCLE(O,A) with $\overline{AD}$ be denoted by O' and refer to Fig. 6. Since the polygonal chain $(A,p_i,\ldots,p_j,D)$ is a convex chain, it is monotonic in the direction of $\overline{AD}$. Now, if p_i were outside DISK(A,C), p_j, by assumption, would have to lie outside DISK(D,B). Therefore $p_i \not\subset$ DISK(O,A) and $p_j \not\subset$ DISK(O,D). It follows that $p_i \subset \mathrm{int}(\Delta DOO')$ and $p_j \subset \mathrm{int}(\Delta AOO')$. Therefore the polygonal chain $(A,p_i,\ldots,p_j,D)$ must intersect $\overline{OO'}$ more than once. Since $\angle OO'A=90°$, this implies that the chain is not monotone in x, a contradiction. Therefore, p_i lies in DISK(A,C) and p_j lies in DISK(D,B) and neither $\overline{AC}$ nor $\overline{DB}$ can be diagonals of the GGD(P), a contradiction.

This completes the proof that no two diagonals can intersect each other or any edge of P and therefore the GGD(P) is planar.

4. ALGORITHMS

In this section we describe two algorithms for computing the RND of a simple polygon P. The algorithms for the GGD are similar. In the straightforward, or brute-force, approach we simply consider each pair of vertices, determine whether they are visible to each other, find all other vertices visible from either of these, and perform the LUNE inclusion tests. This algorithm is presented as Algorithm RND-1 below.

Algorithm RND-1 (brute force)

```
For all pairs of vertices $p_i,p_j$, $i,j=1,2,\ldots,n, i<j$, do the following.

Step 1:  Determine whether $\overline{p_ip_j} \subset P$. If yes, GO TO Step 2 with a
         candidate diagonal;ELSE select a new pair of vertices
         and GO TO Step 1.
```

Step 2: For each vertex p_k of P, k=1,2,...,n, k≠i≠j determine whether at least one of the edges $\overline{p_k p_i}$, $\overline{p_k p_j}$ lies in P. Those that do form the set V_{ij} of vertices visible from either p_i or p_j.

Step 3: For each vertex p_m of V_{ij} determine whether p_m lies in the interior of LUNE(p_i,p_j); if no vertex of V_{ij} lies in the interior of the lune then $\overline{p_i p_j}$ forms a diagonal of the RND; ELSE it does not.

Determining whether an edge lies in P in steps 1 and 2 can be done in O(n) time. Thus Step 1 can be done in O(n) time. In Step 2 O(n) calculations are performed for each of n-2 vertices. Hence Step 2 runs in $O(n^2)$ time for each candidate diagonal determined in Step 1. Finally, Step 3 can be carried out in O(n) time. In the worst case, when P is a convex polygon, $O(n^2)$ candidate diagonals are passed on to Step 2. Thus Algorithm RND-1 is dominated by step 2 and has a worst-case running time of $O(n^4)$.

The *viewability graph* VG(P) of a polygon P is defined by Shamos [13] as a graph the nodes of which represent the vertices of P and that contains an edge between two nodes iff the corresponding vertices in P are visible to each other. Shamos [13] proposed an algorithm for computing VG(P) in $O(n^2)$ time, but El-Gindy [14] has recently discovered that the algorithm fails in some cases. However, El-Gindy and Avis [14] go on to propose an alternative algorithm (with proof of correctness) that does compute VG(P) in $O(n^2)$ time. Algorithm RND-2 below uses the algorithm in [14] for the VG(P) to obtain the RND of P.

Algorithm RND-2

Step 1: Obtain VG(P).

Step 2: For each edge of VG(P), say $\overline{p_i p_j}$, determine whether any neighbors of p_i and p_j in VG(P) lie in the int[LUNE(p_i,p_j)]; if not, then $\overline{p_i p_j}$ forms a diagonal of the RND.

Let e denote the number of edges in VG(P), i.e., e is equal to n plus the number of pairs of vertices in P that are candidate diagonals for the RND(P). Step 1 can be performed in $O(n^2)$ time using the algorithm of El-Gindy and Avis [14]. In Step 2 for each of the e edges of VG(P) at most n-2 neighbors in VG(P) must be tested for lune inclusion. Therefore Step 2 runs in O(ne) time. Clearly e ≤ n(n-1)/2, where equality is attained for convex polygons. A polygon can always be triangulated and a triangulation always contains n-3 diagonals. Furthermore, all triangulations are subgraphs of VG(P). Therefore e ≥ 2n-3, where the equality holds for polygons consisting of two edges $\overline{p_i p_j}$, $\overline{p_j p_k}$ forming an angle less than 180° with a concave polygonal chain joining p_i to p_k. Hence, O(n) ≤ e ≤ $O(n^2)$. Thus Algorithm RND-2 is dominated by Step 2 and has a complexity of O(ne). In the worst-case, for convex polygons, e = $O(n^2)$ and thus the algorithm runs in $O(n^3)$ time. However, in practice this algorithm would most likely be applied to patterns such as those in Fig. 1. In such situations e is closer to O(n) and we can expect the algorithm then to run in less than $O(n^3)$ time.

5. OPEN PROBLEMS

In section 2 it was pointed out that the RNG of a set of points can be computed in O(n log n) time [11]. Furthermore, the RNG of a convex polygon can be computed in O(n) time [11]. In fact, one step of the RNG algorithm in [11] computes the RND defined in this paper in O(n) time. It would thus be surprising if the $O(n^3)$ time for the RND problem of a simple polygon could not be improved. It remains to be shown whether $O(n^3)$ is a lower bound for the problem or whether better algorithms exist. Perhaps the ideas in [11] will help here. A second open problem concerns the components of the RND. From Fig. 1 (b) it is clear that the decomposition tends to yield convex components although not always, as evidenced by the non-convex pentagon at the lower right. It would be nice to find some characterizations of these polygons and relate them to other well known classes of polygons [15].

6. REFERENCES

[1] T. Pavlidis, "Representation of figures by labelled graphs," *Pattern Recognition*, Vol. 4, 1972, pp. 5-17.

[2] B. Chazelle and D. Dobkin, "Decomposing a polygon into its convex parts," *11th ACM Symposium on Theory of Computing*,1979, pp.38-48.

[3] B. Schacter, "Decomposition of polygons into convex sets," *IEEE Transactions on Computers*, Vol. C-27, 1978, pp. 1078-1082.

[4] H. Feng and T. Pavlidis, "Decomposition of polygons into simpler components: Feature generation for syntactic pattern recognition," *IEEE Transactions on Computers*, Vol. C-24, 1975, pp. 636-650.

[5] D. T. Lee and F. P. Preparata, "Location of a point in a planar subdivision and applications," *SIAM J. Computing*, Vol. 6, 1977, pp.594-606.

[6] D. Avis and G. T. Toussaint, "An efficient algorithm for decomposing a polygon into star-shaped polygons," Tech. Rept. No. SOCS 80.8, School of Computer Science, McGill University, May 1980.

[7] G. T. Toussaint, "Pattern recognition and geometrical complexity," *Proc. Fifth International Conference on Pattern Recognition*, Miami Beach, December 1980.

[8] G. T. Toussaint, "The relative neighborhood graph of a finite planar set," *Pattern Recognition*, Vol. 12, 1980, pp. 261-268.

[9] R. B. Urquhart, "Algorithms for computation of relative neighborhood graph," *Electronics Letters*, Vol. 16, July 1980, pp. 556-557.

[10] G. T. Toussaint, "A counter-example to an algorithm for computing the relative neighborhood graph," *Electronics Letters*, to appear.

[11] K. J. Supowit, "The relative neighborhood graph with an application to minimum spanning trees," Tech. Rept. Department of Computer Science, University of Illinois, Urbana-Champaign, August 1980.

[12] D. W. Matula and R. R. Sokal, "Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane,"*Geographical Analysis*, Vol. 12, July 1980, pp. 205-222.

[13] M. I. Shamos, *Problems in Computational Geometry*, Carnegie-Mellon University, 1977.

[14] H. El-Gindy and D. Avis, "A linear algorithm for computing the visibility polygon from a point," Tech. Rept. SOCS 80.9, School of Computer Science, McGill University, Sept. 1980.

[15] G. T. Toussaint, "A hierarchy of simple polygons," manuscript in preparation.

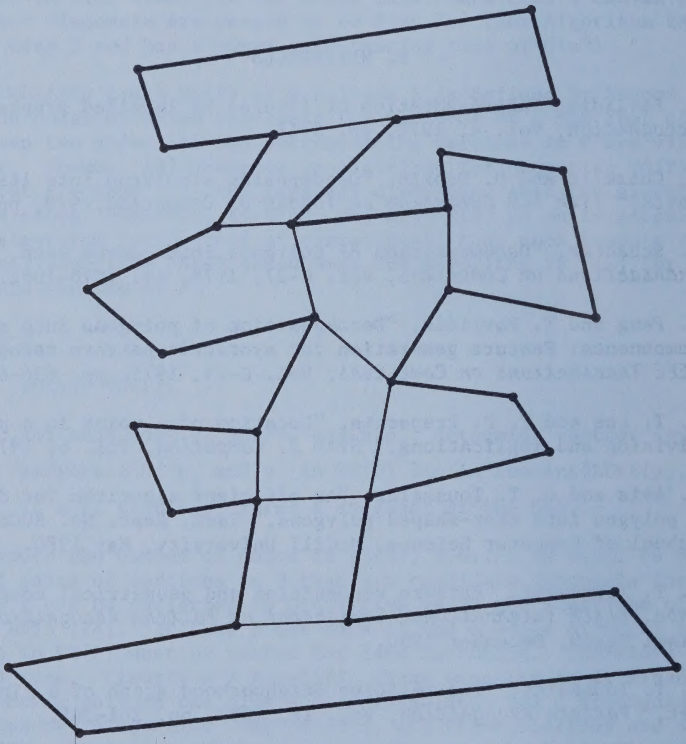


Fig. 7. Perceptually more meaningful decompositions can be obtained by applying some post-processing steps to the basic decompositions. The above decomposition results when we apply a triangle removal operation to the RND of the polygon in Fig. 1 (a). Triangles are "absorbed" into their smallest adjacent component. However such post-processing operations are of no concern in this paper.

PERMUTATION INVERSIONS AND MULTIDIMENSIONAL CUMULATIVE DISTRIBUTION FUNCTIONS

JOHNSON M. HART
Department of Computer Science
University of Kentucky
Lexington, Kentucky 40506

ABSTRACT

The problem of computing the empirical cumulative distribution function (ECDF) of N points in k-dimensional space has been studied and motivated recently by Bentley [1], whose solution uses recursive multidimensional divide-and-conquer. In this paper, the problem is treated as a generalization of the problem of computing the inversion of a permutation. An algorithm of Knuth [2] is then extended to yield an $O(kN(\log N)^{k-1})$ expected time solution to the ECDF problem, which is comparable to Bentley's solution. Neither solution approaches the $O(kN\log_2 N)$ lower bound, and they are worse than the $O(kN^2)$ "brute force" algorithm for large k.

I. INTRODUCTION

Consider a set P of N points in k-space. Point $A=(a_1,a_2,\ldots,a_k)$ is said to dominate point $B=(b_1,b_2,\ldots,b_k)$ iff $a_i>b_i$ (all i, $1\le i\le k$). The rank of A is the number of points in the set which A dominates. Note that this definition of rank is different from that used in [4]. The Empirical Cumulative Distribution Function (ECDF) of P is the rank of each point in P, and the ECDF(N,k) problem is the problem of computing the ECDF for an arbitrary set of N points in k-space. The practical motivation for solving this problem, in terms of statistical and database applications, is given in [1].

The "obvious algorithm" for solving this problem requires that each point be considered individually and compared against all other points. Since each test of the dominance relation can require k comparisons, the time to solve the ECDF(N,k), T(N,k), has an upper bound of

$$O(kN^2).$$

A lower bound on the ECDF is also established easily by extending the usual decision tree argument used for the lower bound on sorting (when k=1, the ECDF problem is the same as sorting). A total of N! different orders are possible in each of the k distinct fields, so the binary decision tree must have at least $(N!)^k$ leaves, and each path from root to leaf of this tree must have

$$O(kN\ \log N)$$

nodes, which yields a lower bound.

Finally, Bentley's divide and conquer algorithm has time behaviour

$$T(N,k) = O(N \lg^{k-1} N),$$

where lg is $\log_2$ and $\lg^{k-1} N$ denotes $(\lg N)^{k-1}$. This solution is an improvement over the "obvious" algorithm for small k and large N, but in certain regions (say k=10, N=1024), Bentley's solution is not as good.

While this paper, unfortunately, does not establish a faster algorithm, it does establish one of comparable complexity, which is much easier to implement and which casts the problem in a different light. The solution presented here was suggested by Vuillemin's paper [3] which followed [1] in the same issue of CACM. These two papers are especially interesting in juxtaposition as they offer considerably different approaches to multidimensional problems.

II. PERMUTATION INVERSIONS AND THE ECDF PROBLEM

Consider the ECDF problem when k=2 so that there are N points in $P=\{p_1,p_2,\ldots,p_N\}$, where each $p_i=(x_i,y_i)$. Without loss of generality, assume that $x_i<x_{i+1}(i=1,\ldots,N)$, or else sort the data so that this property holds. Next, sort the y values to produce a permutation PI so that PI(i) is the rank of y_i with respect to the other y values. That is, y_i is larger than exactly PI(i)-1 of the values in $\{y_1,\ldots y_N\}$. Computing the ECDF of set P, an instance of the ECDF(N,2) problem, is equivalent to computing the function

$$INV(i) = \#\{j \mid i>j \ \& \ PI(i)>PI(j)\}.$$

This function, essentially the rank of the points (i,PI(i)), is called the subdiagonal table of PI by Vuillemin [3], and Knuth [2] calls a minor variation of this function the inversion of the permutation.

Knuth presents two algorithms of time complexity O(N lg N) for computing permutation inversions. One is a variation of two-way merge sort, and no generalization of this is apparent. The other algorithm is given as an answer to Exercise 6 on p. 579, and it is that algorithm which we will generalize here. First, the basic inversion algorithm will be presented in a slightly modified form to produce the function INV as defined above.

Consider each integer value between 0 and N as represented in binary so that bit 0 represents the least significant (right most) bit. For each possible bit (numbered B) (from $\lfloor \lg N \rfloor$ to 0), all of the values of PI(i) (i from 1 to N) are examined in turn with a value of R computed to represent the setting of bit B (0 or 1). S represents the high order bit values; i.e., $S=PI(i)/2^{(B+1)}$. For each B, a running total is kept as i is incremented in array T of the number of PI(j) values for which j<i and PI(j) has bit B reset (R=0) for each possible value of S. Thus, for each i value,

$$T(s) = \#\{j<i \mid S\cdot 2^{(B+1)} \leq PI(j) \leq S\cdot 2^{(B+1)}+2^{B}-1\}.$$

Whenever bit B is set (R=1), simply add the current value of T(s) to the current value of INV(i), for clearly PI(i) will be greater than all those values which contributed to the current value of T(s). Since all bits are examined, all cases will be considered.

Specifically, a pseudo-PL/I program is given below. Only a few comments are given since the preceeding explanation is sufficient.

```
INVERSION_TABLE:
BEGIN;/* ALGORITHM TO COMPUTE THE INVERSION TABLE (INV) OF PERMUTATION
       PI DEFINED ON A SET OF N INTEGERS.  THIS IS A MODIFICATION OF AN
       ALGORITHM GIVEN BY KNUTH.                                       */
       DCL T(0:N/2), INV(N);
       INV=0;  /* INITIALIZE INVERSION TABLE */
BIT_LOOP:DO B=⌊lg N⌋ TO 0 BY -1;
            T=0;  /* INITIALIZE "REGION" COUNTERS */
            DO I=1 TO N;
               R=MOD(⌊PI(I)/2**B⌋,2); /* BIT B OF PI(I) */
               S=⌊PI(I)/2**(B+1)⌋; /* HIGH ORDER BITS */
               IF R=1 THEN INV(I)=INV(I)+T(S);
               IF R=0 THEN T(S)=T(S)+1;
            END;
       END BIT_LOOP;
END INVERSION_TABLE;
```

The running time of this problem is clearly O(N lg N), and since at least one (or possibly two) presort of the initial data is required, we have an O(N lg N) algorithm for the ECDF(N,2) problem. Bentley's algorithm (for the special case of k=2) and Knuth's other algorithm are two additional nontrivial solutions which are equally fast.

III. EXTENSION TO THE MULTIDIMENSIONAL CASE

The extension to higher dimensions is straightforward, except some special techniques will be required to compute the T values in a reasonable amount of space. The solution for k=3 is given here and the generalization to any number of dimensions is evident.

First, assume that N points in 3-space, $P=\{p_1,\ldots,p_N\}$ are given with each $p_i=(x_i,y_i,z_i)$. In addition, assume that the x's are sorted so that $x_i<x_{i+1}$ $(1\leq i<N)$. Now, two permutations, PIY and PIZ, are to be produced

independently on the y and z fields, so that, for instance,

$$\mathrm{PIY}(i) = \#\{j \mid y_i > y_j\}+1.$$

To solve the ECDF(N,3) problem, then, it is necessary to compute the function INV such that

$$\mathrm{INV}(i) = \#\{j \mid i>j \;\&\; \mathrm{PIY}(i)>\mathrm{PIY}(j) \;\&\; \mathrm{PIZ}(i)>\mathrm{PIZ}(j)\}.$$

(Note that one property of INV is lost. When k=2, the value of the original permutation is determined by the inversion table, and this is not the case here.)

In the extended algorithm, T is a two dimensional array and there are two bit numbers, BY and BZ with the property that at any time

$$\mathrm{T}(\mathrm{SY},\mathrm{SZ}) = \#\{j<i \mid \mathrm{SY}\cdot 2^{(\mathrm{BY}+1)} \le \mathrm{PIY}(j) \le \mathrm{SY}\cdot 2^{(\mathrm{BY}+1)}+2^{\mathrm{BY}}-1$$
$$\& \; \mathrm{SZ}\cdot 2^{(\mathrm{BZ}+1)} \le \mathrm{PIZ}(j) \le \mathrm{SZ}\cdot 2^{(\mathrm{BZ}+1)}+2^{\mathrm{BZ}}-1\}.$$

For the time being, ignore the fact that the space requirements for T are excessive (as is the time to initialize it). This problem will be considered after the algorithm.

```
INVERSION_TABLE:
   BEGIN; /*ALGORITHM TO COMPUTE THE MULTIDIMENSIONAL INVERSION TABLE (INV)
           OF PERMUTATIONS PIY AND PIZ (K=3) DEFINED ON A SET OF N INTEGERS.
           THIS IS AN EXTENSION OF THE PERMUTATION INVERSION ALGORITHM.  */
           DCL T(0:N/2,0:N/2), INV(N); /* DIMENSION OF T IS K-1 */
           INV=0;  /* INITIALIZE INVERSION TABLE */
BIT_LOOP_Y:DO BY=⌊lg N⌋ TO 0 BY -1;
  BIT_LOOP_Z:DO BZ=⌊lg N⌋ TO 0 BY -1;
           /* THE NUMBER OF BIT LOOPS IS K-1 */
           T=0; /* INITIALIZE "REGION" COUNTERS */
           DO I=1 TO N;
              RY=MOD(⌊PIY(I)/2**BY⌋,2);/* BIT BY OF PIY(I) */
              RZ=MOD(⌊PIZ(I)/2**BZ⌋,2);/* BIT BZ OF PIZ(I) */
              SY=PIY(I)/2**(BY+1);/* HIGH ORDER BITS OF PIY(I) */
              SZ=PIZ(I)/2**(BZ+1);/* HIGH ORDER BITS OF PIZ(I) */
              IF(RY=1 & RZ=1) THEN INV(I)=INV(I)+T(SY,SZ);
              IF(RY=0 & RZ=0) THEN T(SY,SZ)=T(SY,SZ)+1;
           END;
   END BIT_LOOP_Z;
 END BIT_LOOP_Y;
END INVERSION_TABLE;
```

Now, the only problem with this solution is that the T array requires $O(N^{k-1})$ space and the same amount of time for initialization inside the bit loops. For a given collection of bit numbers (BY,BZ,etc.), however, T is only accessed N times, so it is possible to represent T by some data structure other than an array by regarding the concatenated values (SY,SZ) as a key or index to some data structure. Two solutions come to mind immediately.

a) Balanced Tree. A balanced tree (such as an AVL tree) is used to maintain the T values where (SY,SZ) are concatenated to form an index or lexicographic ordering is used. Making the reasonable assumption that a register can hold a single value (such as SY or SZ), but not more than one, each comparison during a lookup or insertion requires k-1 machine operations. Each reference to T then requires, at worst,

$$O(k \lg N)$$

steps, and the overall algorithm has worst case performance

$$T(N,k) = O(kN \lg^k N).$$

This worst case bound can be refined slightly. Assuming for simplicity that N is a power of 2, then the fraction of the time that T is referenced is $2^{-(k-2)}$, for T is only referenced when all k-1 bits are 0 or all are 1. Taking into account the requirement for the R and S computations, we have

$$\begin{aligned} T(N,k) &= O(kN \lg^{k-1} N) \\ &\quad + O(kN \lg^k N/2^{k-2}) \\ \\ &= O(kN \lg^{k-1} N) + O(kN (\lg N/2)^k) \\ &= O(kN \lg^{k-1} N) + O(kN \lg^k \sqrt{N}). \end{aligned}$$

b) Hash Table. Maintain the T values in a hash table so that each reference to T requires an expected time of O(k) steps (k steps to compute the hash function and make comparisons). Then the algorithm has expected time performance of

$$T(N,k) = O(kN \lg^{k-1} N) \quad \text{(expected)}.$$

These two results are comparable (but not quite as good as) Bentley's algorithm because of the additional factor of k. In both cases, space requirements are no problem. Nonetheless, the algorithm here is quite simple, requiring no recursion and none of the elaborate bookkeeping that is used by Bentley.

Finally, this new routine would work very well with parallel processors. Up to $\lg^{k-1}N$ processors could be employed to represent the different bit values and the pass through the permutation could be achieved in O(kN) time.

REFERENCES

[1] J. L. Bentley, Multidimensional divide-and-conquer, CACM 23, 1980, 214-229.

[2] D. E. Knuth, The Art of Computer Programming, V.3, Sorting and Searching. Addison-Wesley, Reading, Mass., 1973.

[3] J. Vuillemin, A unifying look at data structures, CACM 23, 1980, 229-239.

[4] A. C. Yao and F. F. Yao, On Computing the Rank Function for a Set of Vectors, Technical Report R-75-699, Dept. of Computer Science, University of Illinois, 1975.

LOWER BOUNDS FOR GEOMETRIC PROBLEMS

DAVID AVIS
School of Computer Science, McGill University, Montreal, Quebec H3A 2K6

ABSTRACT

Many lower bounds have been given for 1-dimensional geometric problems under the linear decision tree model. This model is too weak in general to give useful results in higher dimensions. It is shown how to extend such bounds to a decision tree model that includes the primitives usually found in geometric algorithms. A second result is a short proof of a theorem of Yao and Rivest giving a lower bound for the polyhedral decision problem.

1. DECISION TREES

One of the most powerful models for obtaining lower bounds for computational problems is the *decision tree model*. In this model, algorithms are ternary trees, with each internal node representing a test of the form "$f(z_1,\ldots,z_m) : c$". The function f is from a specified class, c is a constant, and $z_1,\ldots,z_m$ are the input numbers. The output of the algorithm takes place at the leaf nodes, and is either a "yes" or "no" answer in the case of a decision problem, or an evaluation of functions from the specified class.

When f is restricted to the class of linear functions, this model is known as the *linear decision tree* model. If the input numbers are real numbers, it is easily seen that the leaf nodes are convex regions of input points. This fact has yielded many lower bounds under the linear decision tree model. The *cost* of a decision tree algorithm is the height of the tree, which is the number of tests required in the worst case.

This paper deals with lower bounds for geometric problems, where the input numbers are usually real numbers of unlimited precision. For example, they may be the co-ordinates of a set of points $x_1,\ldots,x_n$ in k-dimensional euclidean space, R^k. Primitives used in geometric algorithms are usually one of the following three types:

(a) "Is $d_p(x_i,x_j) \geq d_p(x_r,x_s)$?" where i,j,r,s are integers in the range 1 to n and d_p is the p^{th} order Minkowski metric.

(b) "On which side of the hyperplane generated by $\{x_{i_j} : j=1,2,\ldots,k, 1 \leq i_j \leq n\}$ does the point x_r lie?"

(c) "$f(x_1,\ldots,x_n) : c$", where f is a linear function and c is constant.

Tests (a) - (b) can be implemented by linear tests only in 1-dimensional space (ie. k=1). In fact, as was shown in [1], the linear decision tree model is not strong enough even to decide whether three points in the plane are collinear. Since the proof is short, it is included here for completeness.

Consider any linear decision tree algorithm for deciding whether the points (x_1,y_1), (x_2,y_2), and (x_3,y_3) are collinear. Such a tree must decide whether

$$x_3(y_1-y_2)+y_3(x_2-x_1)+y_2x_1-y_1x_2=0 . \tag{1}$$

Since this is a quadratic function, it is always possible to pick collinear points so that $z=(x_1,y_1,x_2,y_2,x_3,y_3)$ does not lie on any of the hyperplanes defined by internal nodes of the decision tree. Therefore z lies in the interior of some 6-dimensional convex set defined by a leaf node of the tree. By perturbing z slightly, we can destroy the collinearity whilst remaining at the same leaf node. Thus the algorithm fails.

When k=1, the linear decision tree model is powerful enough to yield efficient algorithms. Several authors have found sharp linear decision tree bounds, examples of which are given in Section 2. At this stage it is common to make the following statement:

> since the lower bound holds for 1-dimensional problems, it must surely hold for all higher dimensional problems.

Whilst this statement is true, it omits the important fact that the *bound is only true under the linear decision tree model,* and is thus useless in general for $k \geq 2$, as shown above. In Section 3 we attempt to overcome this difficulty by showing that linear decision tree bounds for 1-dimensional problems can be used to give bounds for higher dimensional problems, under a decision tree model that includes the primitives (a)-(c) above. Finally, in Section 4, we give a short proof of a theorem of Yao and Rivest [9] on the complexity of deciding whether a point is contained in a polyhedron.

2. SOME EXAMPLES

We begin by presenting three lower bound results for sets of numbers.

ELEMENT UNIQUENESS: (Dobkin, Lipton [4]) Any linear decision tree algorithm for determining whether a set of n real numbers are distinct requires $\Omega(n \log n)$ tests, in the worst case.

INTERSECTION: (Reingold [7]) Any linear decision tree algorithm for determining whether two sets of n numbers has an empty intersection requires $\Omega(n \log n)$ tests, in the worst case.

ϵ -CLOSENESS: (Fredman, Weide [5]) Any linear decision tree algorithm for determining whether any two of n numbers are within ϵ apart requires $\Omega(n \log n)$ tests, in the worst case.

Consider now, the following set of geometric problems. In each case, it has been shown that one of the preceding bounds is applicable for the 1-dimensional case.

CLOSEST PAIR: (Shamos [8]) Given a set of n points in k-space, determine the pair whose distance is smallest. (Use ELEMENT UNIQUENESS).

FIXED RADIUS NEAREST NEIGHBOUR: (Bentley [2]) Given a set of n points in k-space, and some constant d, list all pairs of points whose distance is no more than d. (Use ϵ - CLOSENESS)

SEGMENT INTERSECTION: (Shamos [8]) Given a set of n line segments in k-space, determine if any pair of segments intersect. (Use ELEMENT UNIQUENESS).

SET DISTANCE: (Bhattacharya [3]) Given two sets of n points in k-space, determine the pair of points (one from each set) whose distance is smallest. (Use INTERSECTION).

Algorithms known to the author for solving the four geometric problems above, use primitives of the form (a)-(c) of Section 1. In the next section we show how to extend the lower bounds above to this class of primitives.

3. LIFTING LOWER BOUNDS TO HIGHER DIMENSIONS

Consider problems where the input is an n-tuple $X_k=(x_1,\ldots,x_n)$ of n points in k-space. Let $f: R^k\times\ldots\times R^k \longrightarrow R$ be a real valued function of nk variables,on the input data. Let $E^k = R \times 0^{k-1}$, be the set of points in R^k that have zeros in every co-ordinate, except possibly the first. Then we say that f is *1-linear,* if the function $f: E^k\times\ldots\times E^k \rightarrow R$ is a linear function. It is easily shown that tests of type (a)-(c) in Section 1 are implementable by 1-linear functions. A *1-linear decision tree* is a decision tree where all tests are by 1-linear functions.

Consider a sequence of geometric problems $\pi = \pi_1,\pi_2,\pi_3 \ldots$ on inputs $X_1,X_2,X_3,\ldots$. π_k is a problem on k-dimensional inputs; its output on X_k is denoted $\pi_k(X_k)$. π is said to be (1-dimensionally) *consistent* if

$$\pi_1(x_1,\ldots,x_n) = \pi_k(y_1,\ldots,y_n) \ \forall(x_1,\ldots,x_n)\epsilon R^n, k=2,3,4,\ldots,$$

where

$$y_i = (x_i, 0, \ldots, 0) \in E^k .$$

It can be verified that the four geometric problems in section 2 are 1-dimensionally consistent. We can now state and prove the following result.

Theorem 1. If π is a consistent sequence and π_1 has a linear decision tree bound of $\Omega(g(n))$, then π_k has a 1-linear decision tree bound of $\Omega(g(n))$.

Proof: Let T_k be a 1-linear decision tree algorithm for π_k. Let f_j be the 1-linear function at some node j of T_k. It follows from the definitions that there exists some function $h: R^n \longrightarrow R$ such that $f(y_1, \ldots, y_n) = h(x_1, \ldots, x_n)$, $\forall (x_1, \ldots, x_n) \in R^n$, where $y_1, \ldots, y_n$ is defined above. Since f is 1-linear, it follows that h is linear. Construct a linear decision tree T_1 by substituting h_j for f_j at each node j of T_k. Since π_k is consistent, it follows that T_1 is a linear decision tree algorithm for π_1, and hence has height at least $\Omega(g(n))$. Thus T_k has height at least $\Omega(g(n))$ and the theorem follows. ☐

Corollary The problems CLOSEST PAIR, FIXED RADIUS NEAREST NEIGHBOUR, SEGMENT INTERSECTION and SET DISTANCE have $\Omega(n \log n)$ 1-linear decision tree bounds.

Thus any improvement in $O(n \log n)$ algorithms for these problems must use primitives other than those listed (a)-(c) in Section 1.

4. THE POLYHEDRAL DECISION PROBLEM

Let $A \in R^{m \times n}$ be an m×n matrix, and let $b \in R^m$ be an m element vector. The set $P = \{x \in R^n : Ax \leq b\}$ is called a *polyhedron*. P is the intersection of m *half-spaces* defined by the m rows of A. Let C(P) denote the minimum height of any linear search tree algorithm T(P) for resolving the decision problem Q(x): "Is $x \in P$?". We call any leaf-node $t \in T(P)$ that gives the output "yes", a *yes-leaf*. For each yes-leaf, t, define the polyhedron, P_t, by

$$P_t = \text{closure } \{x \in R^n : T(P) \text{ resolves } Q(x) \text{ at } t\} .$$

It is easy to construct P_t given T(P). Simply trace the path in T(P) from t to the root, converting any "<" to "≤" and any ">" to "≥". Any equality encountered can be replaced by two inequalities.

We require a few definitions from the field of convex sets, the reader is referred to the book by McMullen and Shephard for further information [6].

A *supporting hyperplane* H of P is a hyperplane such that (i) $H \cap P \neq \emptyset$ and (ii) P is contained in one of the closed halfspaces determined by H. The set $H \cap P$ is then called a *face* F of P. If F has dimension s, then it is called an *s-face*. It is well known that an s-face F can be written as the intersection of n-s of the halfspaces that define P. Thus if the path from a yes-leaf t to the root involves d inequalities, P_t can have at most $\binom{d}{n-s}$ s-faces. The Upper Bound Theorem [6] gives a much sharper bound on this number. We may now state and prove the following result of Yao and Rivest [9].

Theorem 2. $2^{C(P)} \binom{C(P)}{n-s} \geq f_s(P)$.

Proof: Let T(P) be a linear decision tree algorithm for Q(x), and let F be an s-face of P. Since the number of yes-leaves is finite, there must be some yes-leaf t such that $F_t = P_t \cap F$ is an s-dimensional convex set. We will show that F_t is an s-face of P_t. First observe that $P_t \subseteq P$, since each point $y \in P_t$ is the limit of a sequence $\{y_j\}$ of points resolved at the yes-leaf t, and hence in the closed set P. Further $F = P \cap H$, for some supporting hyperplane H. Thus $F_t = P_t \cap H$ and H is a supporting hyperplane for P_t. Hence F_t is an s-face of P_t.

The theorem now follows from the fact that P_t can have at most $\binom{C(P)}{n-s}$ s-faces, and the fact that there are at most $2^{C(P)}$ yes-leaves. □

5. REFERENCES

[1] D. Avis, Comments on a lower bound for convex hull determination, *Information Processing Letters*, to appear.

[2] J. Bentley, Multidimensional divide and conquer, *CACM* 23, 4 (1980).

[3] B. Bhattacharya, The application of computational geometry to pattern recognition problems, *Ph.D. Thesis Proposal*, McGill University, March 1980.

[4] D. Dobkin, R. Lipton and S. Reiss, *Excursions into Geometry*, Computer Science Tech. Rep. 71, Yale University, New Haven, CT. 1976.

[5] M. Fredman and B. Weide, On the complexity of computing the measure $U[a_i, b_i]$, *CACM* 21, 7 (1978) 540-544.

[6] P. McMullen and G. Shephard, *Convex Polytopes and the upper bound conjecture*, Cambridge University Press, 1971.

[7] E. Reingold, On the optimality of some set algorithms, *JACM* 19, 4 (1972) 649-659.

[8] M. Shamos, *Computational Geometry*, Ph.D. Thesis, Yale University (1978).

[9] A. Yao and R. Rivest, On the polyhedral decision problem, *SIAM J. Computing* 9, 2 (1980) 343-347.

An Analysis of Two Heuristics for the Euclidean Traveling Salesman Problem[1]

Jon Louis Bentley
James B. Saxe
Department of Computer Science
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Abstract -- In this paper we study the performance of the *Nearest Neighbor* and *Greedy* heuristics for constructing traveling salesman tours of N cities (points) in the plane. Our first result is negative: we give a particular configuration of cities such that the ratio of heuristic tour length to optimal is $\Theta[(\lg N)/(\lg \lg N)]$, for both heuristics. We then show that if the cities are constrained to lie within the unit square, then both tour lengths are $O(\sqrt{N})$, which implies that the tours are almost surely within a constant factor of optimal if the point sets are randomly drawn from a uniform distribution. Simulation results show that Nearest Neighbor tours are usually very close to optimal. Finally, we discuss several efficient implementations of the Nearest Neighbor heuristic.

1. Introduction

The traveling salesman problem of N cities (points) in the plane calls for constructing a polygon through the cities of minimum perimeter; Papadimitriou [1977] has shown that this problem is NP-complete. Karp [1977] has given an asymptotically efficient algorithm for producing near-optimal tours, but his algorithm is difficult to code and has a high constant factor. In this paper we will investigate the following two heuristics for constructing traveling salesman tours:

- **The Nearest Neighbor Heuristic:** Start at an arbitrary city, and always proceed to the closest unvisited city; when all cities have been visited, return to the starting city.
- **The Greedy Heuristic:** Consider the $\binom{N}{2}$ edges of the graph in increasing order of length, and add each edge to the tour if it will not prohibit the current tour from being extended to a complete tour.

The performance of the Nearest Neighbor heuristic when applied to graphs satisfying the triangle inequality has been studied by Rosenkrantz, Stearns and Lewis [1977]; they showed that its tour is never more than a logarithmic factor from optimal, and gave a family of graphs that exhibit this behavior. They did not, however, consider the performance of the heuristic on graphs induced by point sets, which are a proper subclass of the graphs they studied. In this paper we will study the performance of the above heuristics on cities in the plane.

In Section 2 we discuss the efficacy of the tours produced by the two heuristics. We then turn in Section 3 to efficient implementations of the Nearest Neighbor heuristic. Finally, conclusions are offered in Section 4.

---

[1] This research was supported in part by the Office of Naval Research under contract N00014-76-C-0370.

2. The Performance of the Heuristics

The purpose of this section is to study the performance of the Nearest Neighbor and Greedy heuristics, and we will do this in three distinct ways. In Subsection 2.1 we study the ratio of heuristic tour length to optimal, and show that it can be arbitrarily large. In Subsection 2.2 we then study the performance of the heuristics when confined to point sets in the unit square, and show that it is always good (in a certain precise sense). Finally, in Subsection 2.3, we present data on an actual implementation of the Nearest Neighbor heuristic that shows it performs quite well both on randomly generated data and on "real" data.

2.1. A Bad Point Set

Rosenkrantz, Stearns and Lewis showed that the Nearest Neighbor heuristic always produces a tour of an N-city graph is always within a factor of $(1 + \lg N)/2$ of optimal, and furthermore they displayed a family of "bad" graphs that actually achieve a ratio of $(\lg N)/3$. Unfortunately, the graphs in that family are not embeddable in the plane, and this leaves open the question of whether the heuristic can perform that poorly on planar sets. The following theorem provides a partial answer to that question.

Theorem 2.1:

There exists an infinite family of point sets $\{S_N\}$ such that the set S_N contains N cities, and the tours produced by both the Nearest Neighbor and Greedy heuristics when applied to S_N are longer than the optimal tour by a factor of $\Theta[(\lg N)/(\lg \lg N)]$.

Sketch of Proof:

Let m be any integer greater than 2. Then we construct S_N, where

$$N = m + 1 + \sum_{0 \leq i \leq m} m^i$$

by taking the union of m + 1 sets, $L_0 \cup L_1 \cup \ldots \cup L_m$, where for each non-negative integer $i \leq m$,

$$L_i = \{(jm^{-i}, 2m^{-(i+1)}) \mid 0 \leq j \leq m^i\}.$$

The Nearest Neighbor tour starting at $(0, 2m^{-(m+1)})$ will visit the points in increasing order of their second coordinates, giving a tour length of at least $m + 1$, which exceeds $(\lg N)/(\lg \lg N)$. Other Nearest Neighbor and Greedy tours of S(N) may be slightly shorter but still $\Theta[(\lg N)/(\lg \lg N)]$ (we have omitted the details here). By visiting the points in increasing order of their first coordinates, it is easy to find a tour of length ≤ 6. Thus the claimed ratio is achieved. □

2.2. Point Sets in the Unit Square

In this subsection we will show that if either of the heuristics is applied to a set of N points in the unit square, then the length of the tour is $O(\sqrt{N})$. Indeed, we will show this result for a large class of related heuristics, which we will call *accretive methods* and of which Nearest Neighbor and Greedy are only two examples. We begin by defining a number of terms.

Definition:

Let S be a set of points in the plane. A *partial tour* of S is any graph on S which is a (not necessarily strict) subgraph of some complete tour of S.

Note that any connected component of an incomplete partial tour must be either a path or an isolated point.

Definitions:

Let S be a set of points in the plane. Let T be a partial tour of S and let p and q be points in S. Then,

(1) p is a *terminus* of T if p has degree ≤ 1 in T. In this case, the *coterminal* point to p in T, written coterm(p,T), is defined to be either

  (a) the other terminus of the connected component of T containing p (if p has degree 1 in T) or

  (b) p itself (if p has degree 0 in T).

(2) q is an *eligible neighbor* (with respect to T) of p in S provided that

  (a) {p,q} is not an edge of T and

  (b) The graph obtained by adding the edge {p,q} to T is a partial tour of S.

Note that p has at least one eligible neighbor iff p is a terminus and in this case the eligible neighbors of p are all the termini of T except for p and (if the path containing p does not include all the points of S) coterm(p,T).

Definition:

Let T be a partial tour of a set S of N points in the plane, and let p be a terminus of T. Then we define an *accretion* of T from p to be any partial tour formed by adding to T an edge joining p to a nearest eligible neighbor of p.

Definition:

An approximation method for the traveling salesman problem is an *accretive method* if it produces an approximate tour for a set of N points by constructing a sequence $T_0, T_1, \ldots, T_N$ of partial tours such that

(1) T_0 is the empty partial tour,

(2) For each positive integer $i \leq N$, there exists a point $p_i \in S$ such that T_i is an accretion of T_{i-1} from p_i, and

(3) T_N is the approximate tour.

Any tour T which can be produced by an accretive method is called an *accretive* tour. The sequences $\langle T_i \rangle$ and $\langle p_i \rangle$ together form an *accretive construction*, $(\langle T_i \rangle, \langle p_i \rangle)$, of T. Note that all the p_i need not be distinct; some points in S may be used twice as the starting points for accretion and in this case an equal number will not be so used at all. If each point in in S occurs exactly once in the sequence $\langle p_i \rangle$, then the pair $(\langle T_i \rangle, \langle p_i \rangle)$ is a *directed* accretive construction of T. Any tour which has a directed accretive construction is a *directed* accretive tour.

Note that any tour which can be produced by the Nearest Neighbor heuristic or by the Greedy heuristic must be a directed accretive tour. This result holds trivially for Nearest Neighbor tours. To see that it holds for Greedy tours, consider any Greedy tour T and arbitrarily assign a direction to T, so that each edge has one endpoint distinguished as its *tail* and the other distinguished as its *head* and each vertex is the head of one edge and the tail of another. Then T can be produced by starting with the empty tour and successively accreting from the tails of all edges in increasing order of length.

The key result of this subsection will be that any accretive tour on a set of N points in the unit square must have $O(\sqrt{N})$ length. The main tool in our proof will be the notion of a *compatible circling*, which we now define.

Definition:

Let k be any integer > 2. Then a *k-compatible circling* of the unit square is any set C of circles such that

(1) the center of each circle in C is in the unit square,

(2) at least one point on the circumference of each circle in C is in the unit square, and

(3) there exists no subset of k circles in C such that each one's center is in the interior of the other k–1.

We will be concerned with the maximal values possible for the sums of the radii and of the areas of k-compatible circlings. Our next lemma shows the relevance of sums of radii in k-compatible circlings to the efficacy of accretive methods. Sums of areas turn out to be relevant to efficiency, as we will show in Section 3.

Lemma 2.2:

Let T be any accretive tour on N points in the unit square. Then there exists a 3-compatible circling C of the unit square with N circles such that the sum of the radii of the circles in C is equal to the length of T.

Proof:

Consider any accretive construction $(\langle T_i\rangle,\langle p_i\rangle)$ of T. We define the circling $C = \{c_1, \ldots, c_N\}$ such that for each positive integer $i \leq N$, the circle c_i has p_i as its center and has the distance from p_i to its nearest eligible neighbor with respect to T_{i-1}. We claim that C is a 3-compatible circling of the unit square. To show this, we make the following observations:

(1) For $c_i \in C$, the center of c_i is p_i, which is in the unit square.

(2) Each $c_i \in C$, has at least one point in the unit square on its circumference, namely the nearest eligible neighbor of p_i with respect to T_{i-1}.

(3) For any positive integer $i \leq N$, c_i can have at most one point in its interior which is eligible with respect to T_i, namely $\mathrm{coterm}(p_i,T_{i-1})$. Moreover, accretion from $\mathrm{coterm}(p_i,T_{i-1})$ can occur at most once in the remainder of the construction. Thus no circle in C can have in its interior the center of more than one circle of greater index. Consequently, no three circles in C can all contain each other's centers, since one of the three would have to have the least index.

Thus all the parts of the definition of a 3-compatible circling are satisfied. □

Lemma 2.3:

Let T be any directed accretive tour on N points in the unit square. Then there exists a 2-compatible circling C of the unit square with N circles such that the sum of the radii of the circles in C is equal to the length of T.

Proof:

Use the same construction for C as in the previous proof. Since T must be a directed accretive tour, it follows that for any positive integer $i \leq N$,

$$p^{-1}(\mathrm{coterm}(p_i, T_{i-1})) \leq i,$$

where $p^{-1}(x)$ denotes the unique j such that $p(j) = x$ (which is well defined because T is directed). This means that no circle in C can have in its interior the center of any circle of greater index. Consequently no two circles in C can contain each other's centers. ☐

The following results bound the sum of the radii and areas of the circles in any compatible circling.

Lemma 2.4

Let k be any integer ≥ 2 and let α be any positive real number. Let C be a k-compatible circling of the unit square consisting only of circles with radius $\geq \alpha$. Then,

$$|C| \leq \frac{4(k-1)(1+\alpha^2)}{\pi \alpha^2}.$$

Proof:

For each circle c in C, define the *core* of c to be the circle of radius $\alpha/2$ whose center is the center of c. The cores are all confined to lie entirely within the square of side $1 + \alpha$ formed by extending the original unit square outward by $\alpha/2$ on each side. Since any two circles whose cores overlap must contain each other's centers, no point can be in the interior of more than $k - 1$ of the cores. Thus the sum of the areas of the cores is at most $(k-1)(1+\alpha)^2$. Dividing this area by $\pi\alpha^2/4$ (the area of any core) gives the stated bound. ☐

Theorem 2.5:

Let C be a k-compatible circling of the unit square with N circles. Then,

(1) the sum of the radii of the circles in C is no greater than $\sqrt{4(k-1)N/\pi} + O(\sqrt{k-1}\ \lg N)$ and

(2) the sum of the areas of the circles in C is no greater than $4(k-1)(\ln N) + O(k^{3/2})$.

Proof:

For each positive integer $i \leq N$, let r_i denote the radius of the i-th largest circle in C. By applying Lemma 2.4 to the i largest circles in C, we find that

$$i \leq \frac{4(k-1)(1+r_i^2)}{\pi r_i^2}.$$

Solving for r_i gives

$$r_i \leq \frac{1}{\sqrt{\pi i/4(k-1)} - 1}$$

$$= 2\sqrt{(k-1)/\pi i} + O(k/i).$$

Summing the above over the range $1 \leq i \leq N$ gives the bound of part (1). If we let A_i be the area of the i-th largest circle, we find

$$A_i = \pi r_i^2 \leq 4(k-1)/i + O((k/i)^{3/2}).$$

Summing this gives the bound stated in part (2). ☐

Using Lemmas 2.2 and 2.3, we may interpret part (1) of this result in the domain of accretive tours, giving the following corollaries.

Corollary 2.6:

The length of any accretive tour on a set S of points in the unit square is bounded above by $4\sqrt{2N/\pi} + O(\lg N)$, where N is the cardinality of S.

Corollary 2.7:

The length of any directed accretive tour on a set S of points in the unit square is bounded above by $4\sqrt{N/\pi} + O(\lg N)$, where N is the cardinality of S.

Since tours produced by the Greedy and Nearest Neighbor heuristics are always directed accretive tours, the length of any such tour on a set of points in the unit square is bounded by Corollary 2.7.

A well-known theorem of Beardwood, Halton and Hammerslee [1959] states that if N points are drawn uniformly on the unit square, then the length of their traveling salesman tour is almost surely proportional to $\sqrt{N}$; thus our result implies that both the Greedy and Nearest Neighbor tours are almost always within a constant factor of optimal for such sets. Another interpretation of this result is that although the heuristic tour can be a logarithmic factor longer than the optimal, this can happen only because the optimal tour is inordinately short, and not because the heuristic tour is long.

The constant factors on the bounds above are subject to some improvement. For example, if we take advantage of the fact that the maximum packing density of circles in the plane is $\pi/2\sqrt{3}$, the bound on the length of any directed accretive tour can be improved from $4\sqrt{N/\pi}$ (aproximately $2.26\sqrt{N}$) plus lower order terms to $2\sqrt{2N}/\sqrt[4]{3}$ (approximately $2.15\sqrt{N}$) plus lower order terms. A point set whose optimal tour is at least half this great, minus lower order terms, can be constructed by taking the centers of N equal circles of the maximum size that can be packed in the unit square without overlap. It is also interesting to note that the preceding analysis can be extended to d-dimensional space for $d \geq 2$, giving a bound on the ratio of worst-case directed accretive tour length to worst-case optimal tour length which approaches d/(d-1) for large point sets in the d-cube.

2.3. Data on the Efficacy of the Nearest Neighbor Heuristic

In the previous subsection we investigated a number of heuristics and showed that the worst-case length of their respective tours are within a constant factor of the expected length of the tour of points uniform in the unit square. In this subsection we will briefly study the performance of one of those heuristics--Nearest Neighbor--from several other viewpoints. As a comparison, we also implemented the Minimum Spanning Tree heuristic of Rosenkrantz, Stearns and Lewis [1977] that produces an approximate tour by traversing the minimum spanning tree of the point set.

The first set of data we gathered dealt with the performance of the heuristics on point sets randomly generated on the unit square. The Nearest Neighbor heuristic tended to produce tours with length $.92\sqrt{N}$ while the MST heuristic produced tours of length $.95\sqrt{N}$. By way of comparison, the length of the minimum spanning tree was usually $.66\sqrt{N}$; Papadimitriou cites a reference asserting that the length of optimal tours is usually close to $.765\sqrt{N}$ (that is, .765 was an empirical estimate of the constant defined by Beardwood, Halton and Hammerslee). Thus we see that on uniform data, the Nearest Neighbor heuristic finds tours that are only 20% longer than optimal and consistently produces better tours than the Minimum Spanning Tree heuristic. We also tested point sets with coordinates drawn independently from normal distributions; the relative performance of the two heuristics remained unchanged.

We also applied the two heuristics to "real data" that arose in an application of the planar traveling salesman problem. The application called for scheduling the movements of a mechanical plotter to plot a set of points representing precincts in a geopolitical database. We investigated two sets of data: the first contained 318 points representing precincts in the city of San Diego which were fairly uniformly distributed; the second set contained 1122 points from the state of New Mexico which were highly clustered. The mean tour length for San Diego was 14.74 for the Minimum Spanning Tree heuristic and 15.19 for the Nearest Neighbor heuristic; for New Mexico the performances were 16.57 for Minimum Spanning Tree and 14.88 for Nearest Neighbor. Thus the Nearest Neighbor heuristic performed slightly worse than the Minimum Spanning Tree heuristic on the uniform data and much better on the highly clustered data.

The authors feel that the data of this brief subsection together with the analysis of the previous subsection shows that the Nearest Neighbor heuristic produces tours that are "good enough" for many applications. Furthermore, the Nearest Neighbor heuristic appears to be quite competitive with the Minimum Spanning Tree heuristic, and is much easier to implement.

3. Implementing the Nearest Neighbor Heuristic

In this section we will study the problem of efficiently implementing the Nearest Neighbor heuristic for a planar point set. The most straightforward implementation, of course, finds the nearest neighbor to the current city by examining all unvisited cities. Although this method has running time of $\Theta(N^2)$, it can still be practical for many applications because of the simplicity of the code (less than a dozen lines of Pascal suffice) and the small constant of proportionality. We will now examine a set of algorithms for implementing the heuristic that are asymptotically more efficient. Due to the space limitations of these proceedings, however, we can only state the main theorems and then briefly sketch their proofs.

The first theorem that we will investigate provides an upper bound on the worst-case complexity of the Nearest Neighbor heuristic; it is based on a data structure of Bentley and Saxe [1980].

Theorem 3.1:

The Nearest Neighbor tour of N points in the plane can be found in worst-case time of $O(N^{3/2} \lg N)$.

Sketch of Proof:

The algorithm repeatedly finds the nearest neighbor of the most recently visited city among the constantly decreasing set of unvisited cities. We employ the data structure for nearest neighbor searching of Bentley and Saxe [1980]; they showed that it can be initialized in $O(N \lg N)$ time and that both searches and deletions can be performed in $O(N^{1/2} \lg N)$ time. □

Reducing the worst-case complexity of implementing the Nearest Neighbor heuristic remains an open problem.

We turn now to the expected complexity of algorithms that implement the Nearest Neighbor heuristic. Specifically, we shall assume that the point sets are uniformly distributed in the unit square. The first expected-time theorem slightly improves the worst-case bound of Theorem 3.1.

Theorem 3.2:

The expected running time of a projection algorithm to construct the Nearest Neighbor tour of N points distributed uniformly in the unit square is $\Theta(N^{3/2})$.

Sketch of Proof:

By a projection algorithm we mean one that accomplishes nearest neighbor searching by projecting the points onto the x-axis and solving a resulting one-dimensional problem; for details on this method, see Papadimitriou and Bentley [1980]. We use three important lemmas in the analysis of this method, two of which relate to dividing the unit square into $\sqrt{N}$ equal-sized vertical slabs (these slabs are for the sake of analysis only and are not present in the algorithm). (1) The length of the tour is bounded above by $O(\sqrt{N})$. (2) By (1), the tour never visits more than $\Theta(N)$ slabs, and projection searching never examines more than $\Theta(N)$ slabs. (3) With extremely high probability, no slab contains more than $2\sqrt{N}$ points. We then use a technique that Knuth [1980] refers to as "oh-of-probability" analysis to show that these facts together establish the theorem. □

Although this theorem produces only a slightly better asymptotic result than Theorem 3.1, its algorithm is much easier to implement and should prove quite practical in many contexts.

As a final expected time algorithm, we present the following.

Theorem 3.3:

The expected running time of a cell-based algorithm to construct the Nearest Neighbor tour of N points distributed uniformly in the unit square is $O(N \lg^2 N)$.

Sketch of Proof:

The cell-based algorithm divides the unit square into N nonoverlapping squares and uses those to find the nearest neighbors. The proof of the running time depends on two crucial lemmas. First, with extremely high probability, no square contains more than $\lg N$ points. Second, the bound on the sum of the areas of the circles in a 2-compatible circling can be used to show that at most $O(N \lg N)$ cells will be examined in the search. These facts can be combined by Knuth's [1980] method to yield the claimed theorem. □

4. Conclusions

This paper contains two distinct kinds of contributions. On a theoretical level, we have analyzed the performance of two important heuristics--Nearest Neighbor and Greedy--when applied to planar problems. We showed first that the ratios of heuristic tour length to optimal can be almost $\Theta(\lg N)$, and then showed that this occurs not because the heuristic is extremely long but rather because the optimal is extremely short. The analytic method we used for the latter upper bound is quite general and powerful; we showed that it can be applied to a broad class of heuristics. On a more practical level, the experimental evidence of Subsection 2.3 shows that the Nearest Neighbor heuristic usually produces very good tours, and Section 3 shows that that heuristic can be implemented to run very quickly.

References

Beardwood, J., J. H. Halton and J. M. Hammersley [1959]. "The shortest path through many points," *Proceedings of the Cambridge Philosophical Society 55*, pp. 299-327.

Bentley, J. L. and J. B. Saxe [1980]. "Decomposable searching problems I: Static-to-dynamic transformations," to appear in the *Journal of Algorithms 1*, 4, December 1980.

Karp, R. M. [1977]. "Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane," *Mathematics of Operations Research 2*, 3, pp. 209-224.

Knuth, D. E. [1980]. Lecture delivered in the class Computer Science 255, Stanford University, Spring Quarter 1980.

Papadimitriou, C. H. [1977]. "The euclidean traveling salesman problem is NP-complete," *Theoretical Computer Science 4*, pp. 237-244.

Papadimitriou, C. H. and J. L. Bentley [1980]. "A worst-case analysis of nearest-neighbor searching by projection," *Seventh Inter- national Conference on Automata, Languages and Programming*, pp. 470-482, August 1980, Springer-Verlag. (Also available in Carnegie-Mellon Computer Science Report CMU-CS-80-109.)

Rosenkrantz, D. J., R. E. Stearns, and P. M. Lewis [1977]. "An analysis of several heuristics for the traveling salesman problem," *SIAM J. Computing 6*, 3, pp. 563-581.

ON EXTENDED BCH CODES

RICHARD E. BLAHUT
International Business Machines, Owego, New York 13827

A class of extended BCH Codes will be constructed from the class of extended Reed-Soloman codes. This class will be found to contain many of the best codes known, and one new code better than any previously known.

I. Introduction

The best codes known, as compiled by MacWilliams and Sloane [1] in their Appendix A, are often either BCH codes, Goppa codes, or codes in the classes obtained by Sloane, Reddy and Chen [2] and by Kasahara, Sugiyama, Hirasawa, and Namekawa [3]. However, the constructions in [2] and [3] are technical, non-intuitive and forbidding to most potential users. In this paper we will give a simple construction that unifies the codes of [2] and [3] and also allows the construction of some new codes. We will give one example of a new code better than the entries of MacWilliams and Sloane's Appendix A.

The construction will extract the codes of [2] and [3] from extended Reed-Solomon codes. This compliments the observation of Delsarte [4] that Goppa codes can be obtained from modified Reed-Solomon codes. Consequently, a universal Reed-Solomon decoder can be built that, with a few switchable options, can decode many of the best codes known.

II. Extended Reed-Solomon Codes

It is well-known that a Reed-Solomon code can be extended by two extra information symbols without reducing the minimum distance [5]. We will place one new symbol at the beginning and one at the end of the Reed-Solomon codeword. The codewords are written as;

$$(c_-, c_0, c_1, c_2, \ldots, c_{q^m-3}, c_{q^m-2}, c_+)$$

and $n = q^m+1$. The vector obtained by excluding c_- and c_+ will be called the interior. We shall study extended codes by means of the Fourier transform spectrum of the interior, defined as;

$$C_j = \sum_{i=0}^{n-3} \alpha^{ij} c_i \qquad j = 0, \ldots, n-3$$

with α primitive, together with additional properties of the extended vector space. When we speak of the spectrum of the code word, we mean the spectrum of the interior.

Definition

Let j_0 and t be arbitrary integers. A $(q + 1, q + 1-2t, 2t + 1)$ extended Reed-Soloman code is the linear code over GF(q) consisting of all words $(c_-, c_0, c_1, \ldots, c_{q-1}, c_+)$ that have spectra satisfying;

i) $C_j = 0 \qquad j = j_0 + 1, \ldots, j_0 + 2t-2$

ii) $C_{j_0} = c_-$

iii) $C_{j_0 + 2t-1} = c_+$

The integer $2t + 1$ is the designed distance of the extended Reed-Solomon code. The definition constrains $2t-2$ successive spectral components to be equal to zero, and the spectral components on either side of these $2t-2$ are equal to c_- and c_+ respectively. The two special parity frequencies j_0 and $j_0 + 2t-1$ will be called the edge frequencies.

There are $q + 1 - 2t$ components of the spectrum that can take on arbitrary values. It will be convenient to think of these $q + 1 - 2t$ spectral components as representing the $q + 1 - 2t$ information symbols. We call this encoding in the frequency domain. A more detailed description of Reed-Solomon codes using a spectral terminology can be found in [6] and [7].

III. Extended BCH Codes

Now consider codes for the channel alphabet GF(q) construced from Reed-Solomon codes over $GF(q^m)$. Recall that a BCH code over GF(q) can be obtained from a Reed-Solomon code over GF(q^m) by selecting all the codewords that have components only in GF(q). Similarly, an extended Reed-Solomon code over GF(q^m) can be used to create a code over GF(q) by taking the subfield-subcode of the Reed-Solomon code in the smaller field GF(q). Sometimes, because of constraints over the subfield GF(q), the extended components might not be useable as information places. That is, they might contain the same symbol in every codeword, and may just as well be dropped.

In this section, we shall take a somewhat different approach to constructing a code over GF(q) from an extended code over GF(q^m). The approach is to take from the extended Reed-Solomon code those words that have components only in GF(q) in the q^m-1 interior symbols. The two extended symbols, c_- and c_+, are allowed to be arbitrary in GF(q^m), but are represented in the codeword by leading and trailing "tails" each consisting of m q-ary symbols.

As we shall see, it takes a little effort to make this work, but the resulting codes can be decoded by a Reed-Solomon decoder, and many of the best known codes can be obtained this way. We call these codes extended BCH codes. One can find the original BCH code in the extended BCH code by picking all codewords whose tails are equal to zero.

An encoder in the frequency domain must insert information into the spectrum in only those q^k ways that will only give rise to q-ary components in the time domain. The valid spectra must satisfy certain constraints. In particular, the vector C_j over GF(q^m) of length q^m-1 a valid spectrum for the interior of a GF(q)-ary codeword if the conjugacy constraints

$$C_j^q = C_{qj}$$

are satisfied. For example, in Figure 1 are shown the conjugacy classes for GF(64). In each class are the values of the index j that are locked together by the conjugacy constraint. For each class, only one component of the spectrum can be independently specified.

To get an extended BCH code, we proceed as for the extended Reed-Solomon

0
1, 2, 4, 8, 16, 32
3, 6, 12, 24, 48 33
5, 10, 20, 40, 17, 34
7, 14, 28, 56, 49, 35
9, 18, 36
11, 22, 44, 25, 50, 37
13, 26, 52, 41, 19, 38
15, 30, 60, 57, 51, 39
21, 42
23, 46, 29, 58, 53, 43
27, 54, 45
31, 62, 61, 59, 55, 47

Conjugacy Classes for GF (64)

Figure 1

codes but also require the constraints be satisfied. Choose 2t contiguous frequencies as parity frequencies and set

$$C_{j_0} = c_-$$

$$C_j = 0 \qquad j = j_0 + 1, \ldots, j_0 + 2t-2$$

$$C_{j_0 + 2t-1} = c_+$$

where c_- and c_+ are arbitrary elements of $GF(q^m)$ provided they do not violate the conjugacy constraints. The remining spectral components are chosen in any of the q^k ways that satisfy the conjugacy constraints. The interior of the codeword is then the inverse Fourier transform of this spectrum. The interior is preceeded by a tail consisting of at most m GF(q)-ary symbols representing c_-, and is followed by similar tail representing c_+. It may be that constraints on C_{j_0} or C_{j_0+2t-1} force c_- or c_+ to have an order smaller than q^m-1 in which case c_- or c_+ represents less than m q-ary information symbols. In such a case, the tail length is a divisor of m.

In contrast to BCH codes, where the choice of $j_0 = 1$ will usually give the best codes, we will find good extended BCH codes usually by choosing $j_0 = 0$ or -1, but sometimes other values for j_0 will prove worthwhile.

First, take $j_0 = 0$. All of the extended codes obtained from primitive BCH codes of blocklength $n \leq 255$ are shown in Figure 2. The BCH codes that are extended have parity frequencies $C_0, \ldots, C_{2t-1}$, but for comparison purposes, we give the parameters of the primitive BCH code with parity frequencies $C_1, \ldots, C_{2t}$. As an example of the construction, we construct some codes with n = 63 by using the conjugacy classes shown in Figure 1. Since $C_0^2 = C_0$, it must be either zero or one, and c_- is a single binary symbol. C_{2t-1} and hence c_+ is an arbitrary element of GF(64) unless 2t-1 = 9, 18, 36, 27, 54, or 45 in which case c_+ contains three bits, or 2t-1 = 21 or 42 in which case c_+ contains two bits. Further, 2t-1 must be the smallest integer in its conjugacy class because if there is another integer in its conjugacy class, this component of the spectrum in constrained to zero and so C_{2t-1} must also be zero.

Let $t = 4$. Then $2t-1 = 7$ and $C_0, C_1, \ldots, C_7$ are parity frequencies with $C_0 = c_-$ any element of GF(2) and $C_7 = c_+$ any element of GF(64). Further, C_0 corresponds to one parity bit, and C_1, C_3, C_5, and C_7 each corresponds to six parity bits. Hence $n = 63 + 1 + 6$ and we have a (70, 45) code whose minimum distance is at least 9.

Let $t = 8$. Then $2t-1 = 17$ and $C_0, C_1, \ldots, C_{17}$ are parity frequencies with $C_0 = c_-$ any element of GF(2) and $C_{17} = c_+$ any element of GF(64). Further, C_0 corresponds to one parity bit, C_1, C_3, C_5, C_7, C_{11}, C_{13}, C_{15}, and C_{17} each corresponds to six parity bits, and C_9 corresponds to three parity bits. Hence we have a (70, 24) code whose minimum distance is at least 17.

This second example illustrates a case where the extended BCH code has a designed distance d that is smaller than the minimum distance of the original BCH code. This occurs several times in Figure 2 whenever the BCH code of designed distance d is the same as the BCH code of designed distance d + 2.

All of these codes can be decoded as extended Reed-Soloman codes. One simple description is to decode twice. First decode with the assumption that there are no errors in the tails, and hence there are 2t syndromes available to correct t errors in the interior. Then decode with the assumption that there are at most t-1 errors in the interior. Then there are 2t-1 syndromes available to first correct the interior and later correct the edge symbols. Only one of these trial decodings can be successful because the minimum distance of the code is known to be at least 2t + 1. In practice, the two trial decodings can be combined into a single decoder.

****

Next, take $j_0 = -1$. With this choice of j_0, the minimum distance of the code is even. To make the minimum distance odd, we add one extra bit as a parity check on the bits of one of the tails. The parity bit allows the decoder to detect a single error in this tail. The extra information allows correcting an extra error. Hence the minimum distance in increased by one. All of the extended codes obtained from primitive BCH codes of blocklength n = 255 are shown in Figure 3.

Again, all of these codes can be decoded as Extended Reed-Solomon codes. Make these trial decodings under the hypothesis that the interior has t-2 or less errors; that it has exactly t-1 errors; and that it has exactly t errors. In the first case, 2t-3 syndromes are available to correct the interior and then the edge symbols. In the second case, one edge symbol has a single error, and the extra parity bit identifies which it is. Using the other edge syndrome, 2t-1 syndromes are available to correct the interior and then the other edge symbol. In the third case, 2t + 1 syndromes are available to correct the interior. Only one of these trial decodings can be successful.

****

We can also take other values for j_0, but we will find few codes of interest in the range $n \leq 500$, $d \leq 29$. The only exception is in GF(64). With $j_0 = -9$ one can obtain a (70, 20, 21) code as an extended BCH code which is better than any other known code of this n and k.

d_{BCH}	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k
3	(15,11)	(20,15)	5	(31,26)	(37,31)	6	(63,57)	(70,63)	7
5	(15,7)	(20,11)	9	(31,21)	(37,26)	11	(63,51)	(70,57)	13
7	(15,5)	(18,7)	11	(31,16)	(37,21)	16	(63,45)	(70,51)	19
9					(37,16)	21	(63,39)	(70,45)	25
11				(31,11)			(63,36)	(67,39)	28
13					(37,11)	26	(63,30)	(70,36)	34
15				(31,6)			(63,24)	(70,30)	40
17								(70,24)	46
19									
21							(63,18)		
23							(63,16)	(66,18)	48
25								(70,16)	54
27							(63,10)		
29									

d_{BCH}	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k
3	(127,120)	(135,127)	8	(255,247)	(264,255)	9
5	(127,113)	(135,120)	15	(255,239)	(264,247)	17
7	(127,106)	(135,113)	22	(255,231)	(264,239)	25
9	(127,99)	(135,106)	29	(255,223)	(264,231)	33
11	(127,92)	(135,99)	36	(255,215)	(264,223)	41
13	(127,85)	(135,92)	43	(255,207)	(264,215)	49
15	(127,78)	(135,85)	50	(255,199)	(264,207)	57
17		(135,78)	57	(255,191)	(264,199)	65
19	(127,71)			(255,187)	(260,191)	69
21	(127,64)	(135,71)	64	(255,179)	(264,187)	77
23	(127,57)	(135,64)	71	(255,171)	(264,179)	85
25		(135,57)	78	(255,163)	(264,171)	93
27	(127,50)			(255,155)	(264,163)	101
29	(127,43)	(135,50)	85	(255,147)	(264,155)	109

TABLE OF EXTENDED BCH CODES

$j_o=0$

FIGURE 2

d_{BCH}	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k
3	(15,11)			(31,26)			(63,57)		
5	(15,7)	(24,14)	10	(31,21)	(42,30)	12	(63,51)	(76,62)	14
7	(15,5)	(24,10)	14	(31,16)	(42,25)	17	(63,45)	(76,56)	20
9					(42,20)	22	(63,39)	(76,50)	26
11				(31,11)	(42,15)	27	(63,36)	(76,44)	32
13							(63,30)	(73,38)	35
15				(31,6)	(42,10)	32	(63,24)	(76,35)	41
17								(76,29)	47
19								(76,23)	53
21							(63,18)		
23							(63,16)		
25								(72,17)	55
27							(63,10)	(76,15)	61
29									

d_{BCH}	PRIMITIVE BCH CODE	EXTENDED BCH	n-k	PRIMITIVE BCH CODE	EXTENDED BCH	n-k
3	(127,120)			(255,247)		
5	(127,113)	(142,126)	16	(255,239)	(272,254)	18
7	(127,106)	(142,119)	23	(255,231)	(272,246)	26
9	(127,99)	(142,112)	30	(255,223)	(272,238)	34
11	(127,92)	(142,105)	37	(255,215)	(272,230)	42
13	(127,85)	(142,98)	44	(255,207)	(272,222)	50
15	(127,78)	(142,91)	51	(255,199)	(272,214)	58
17		(142,84)	58	(255,191)	(272,206)	66
19	(127,71)	(142,77)	65	(255,187)	(272,198)	74
21	(127,64)			(255,179)	(268,190)	78
23	(127,57)	(142,70)	72	(255,171)	(272,186)	86
25		(142,63)	79	(255,163)	(272,178)	94
27	(127,50)	(142,56)	86	(255,155)	(272,170)	102
29	(127,43)	(142,49)	93	(255,147)	(272,162)	110

TABLE OF EXTENDED BCH CODES

$j_o=1$

FIGURE 3

References

1. MacWilliams, F. J. and Sloane, N. J. A., "The Theory of Error Correcting Codes," North Holland, Amsterdam, 1977.

2. Sloane, N. J. A., S. M. Reddy, and C. L. Chen, "New Binary Codes", IEEE Trans. Information Theory, Vol 19 pp 503-510, 1972.

3. Kasahara, M., Y. Sugiyama, S. Hirasawa, and T. Namekawa, "A New Class of Binary Codes Constructed on the Basis of BCH Codes", IEEE Trans. On Information Theory, Vol 21 pp 582, 585, 1975.

4. Delsarte, P., "On Subfiled Subcodes of Modified Reed-Solomon Codes", IEEE Trans. Information Theory, Vol. IT 21, pp 575-576, September 1975.

5. Wolf, J. K., "Adding Two Information Symbols to Certain Nonbinary BCH Codes and Some Applications", Bell System Technology J., pp 2405-2424, 1969.

6. Blahut, R. E., "Transform Techniques for Error Control Codes", IBM Journal of Research and Development Vol. 23, pp 299-315, 1979.

7. Blahut, R. E. "Algebraic Decoding in the Frequency Domain" in Algebraic Coding Theory and Practice, G. Longo editor Springer-Verlag, 1979.

REDUCED-STATE SOFT-DECISION TRELLIS DECODING OF LINEAR BLOCK CODES[†]

K. R. MATIS and J. W. MODESTINO
Electrical and Systems Engineering Department
Rensselaer Polytechnic Institute
Troy, New York 12181

ABSTRACT

An algorithm for soft-decision decoding of linear block codes is presented. This algorithm performs a reduced-state search through a trellis derived from the parity check matrix of an (n,k) linear block code. The computational complexity of the algorithm is considerably reduced from that of a full maximum likelihood algorithm. We demonstrate the tradeoff between complexity and efficiency of the algorithms through computer simulation.

I. INTRODUCTION

Several authors have devised decoding schemes for linear block codes which utilize the channel measurement information contained in a real-valued decision statistic available at the output of a receiver [1]-[6]. However, most of these algorithms are either unreasonably complex or are applicable only to restricted classes of codes.

Wolf [6] has proposed an algorithm which has an interesting representation as a search for the maximum likelihood path through a trellis which can be defined for a block code. This scheme utilizes the Viterbi algorithm (VA) in implementing the search and has been shown to perform maximum likelihood decoding. Unfortunately, the computational complexity of the VA increases with the width of the trellis to be searched. In its application to the decoding of convolutional codes, the VA becomes computationally unfeasable for codes of constraint length greater than 10 or 12. There is a similar limitation when the VA is used for maximum likelihood decoding of linear (n,k) block codes over GF(q). The width of the trellis in this case can be as high as $q^{(n-k)}$ and the VA is too complex for many codes of even moderate length.

This paper is concerned with the development of a reduced-state algorithm which is essentially a generalization of the VA. The complexity of this algorithm is a function of several parameters and we demonstrate that the algorithm gives performance close to the VA when applied to the block code trellis defined by Wolf at the same time the complexity is significantly reduced.

II. TRELLIS DECODING OF LINEAR BLOCK CODES

When applied to the maximum likelihood decoding of a binary (n,k) linear block code, the VA is used to search a trellis which is defined in terms of the parity check equations satisfied by the 2^k code words of the code. The decoder attempts to track the "state" of the partial parity check sum:

$$\underline{S}(k) = \sum_{i=1}^{k} z_i \, \underline{h}_i \; , \quad k=0,1,\ldots,n \quad .$$

[†] This work was supported in part by ONR under Contract No. N00014-75-C-0281, and in part by USAF RADC under Contract No. AF30602-78-C-0083.

Here $\underline{h}_i$ represents the i'th column of the (n-k) x n parity check matrix $\underline{\underline{H}}$ and z_i is the i'th entry in a binary row vector $\underline{z}$. To each sequence z_i, $i=1,2,\ldots,n$ there corresponds a sequence of parity check sums $\underline{S}(k)$, $k=1,2,\ldots,n$.

We can use a trellis diagram to represent the evolution in time of this sequence of parity check sums. The partial parity check sum can be considered a state so that the trellis diagrams the transitions between these states from one node level to the next.

A trellis for a (5,3) code with parity check matrix

$$\underline{\underline{H}} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix} ,$$

is shown in Fig. 1. Each binary n-tuple traces a distinct path through the trellis corresponding to a sequence of transitions of the parity check states.

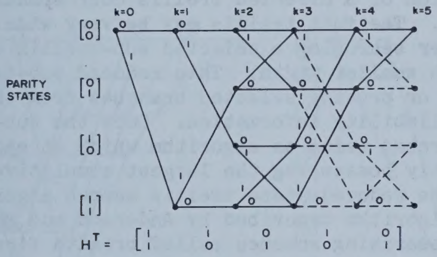

k=0
k=1
k=2
k=3
k=4
k=5
PARITY STATES
$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 0 \end{bmatrix}$
$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$
$H^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}$

Figure 1

Trellis for Binary (5,3) Code

We know that for each of the 2^k binary sequences of length n which are in the code the corresponding sequence of parity check states must end with the all zeros state. That is, $\underline{S}(n) = \underline{0}$ since

$$\underline{x}\,\underline{\underline{H}}^T = \sum_{i=1}^{n} x_i\,\underline{h}_i = \underline{0} ,$$

if $\underline{x}$ is a codeword.

Clearly, then, a path through the trellis corresponding to a codeword must terminate at the all zero node at the final level. The dotted branches in Fig. 1 can then be eliminated from consideration.

The VA operates by finding the path through the code trellis, corresponding to a codeword, which maximizes the log-likelihood functional

$$L_m\{\underline{r}\} = \log f_{\underline{r}|\underline{x}_m}(\underline{r}|\underline{x}_m) ,$$

where $\underline{r}$ represents the real-valued receiver output sequence, and $f_{\underline{r}|\underline{x}_m}(\cdot|\underline{x}_m)$ describes the channel transition probability mechanism. If we assume that the channel affects each transmitted symbol independently, the log-likelihood functional becomes

$$L_m\{\underline{r}\} = \sum_{i=1}^{n} \log f(r_i|x_{mi}) = \sum_{i=1}^{n} d_i .$$

In the particular case of matched filter reception of binary antipodal signals transmitted over an additive white Gaussian noise (AWGN) channel, the receiver output sequence is of the form:

$$r_i = \begin{cases} \sqrt{2E_s/N_0} + n_i \; ; & x_i=0 \\ -\sqrt{2E_s/N_0} + n_i \; ; & x_i=1 \end{cases}$$

where n_i is a zero-mean Gaussian variate possessing unit variance. The branch metric corresponding to the i'th level along the path representing code vector $\underline{x}_m$ can then be computed as

$$\partial_i = r_i(-1)^{x_{mi}} \quad , \quad i=1,2,\ldots,n \quad .$$

The cumulative path metric is simply the sum of the branch metrics along a specified path up to a given level in the trellis.

III. A TRELLIS SEARCHING ALGORITHM

We consider a section of a directed trellis corresponding to a binary (n,k) linear block code. The full trellis may be very wide. In general, however, we will consider searching a selected sub-trellis of the full trellis which has a much smaller width. This reduced sub-trellis can be obtained by eliminating or pruning selected branches from the trellis on the basis of channel reliability information. Once the sub-trellis has been defined, it is searched using an algorithm which at each level extends only the M paths currently possessing the largest cumulative path metric. This latter aspect of the reduced-state trellis search algorithm has some similarity to the M-L algorithm described by Anderson and Mohan [8]. It is one of a class of tree searching schemes called breadth first algorithms. If $M=q^{n-k}$ (we do not purge any nodes at any node level) and we extend each of the branches emanating from the nodes, this algorithm is equivalent to the Viterbi algorithm and will perform true maximum likelihood decoding. We now demonstrate how the channel reliability information can be used to determine an appropriate subset of the branches to be extended from each node.

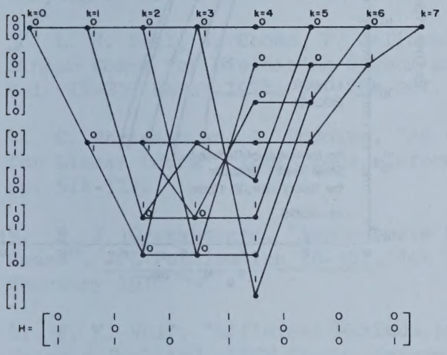
Let the received real-valued n-vector be represented by $\underline{r}=(r_1,r_2,\ldots,r_n)$. We also associate a real-valued vector $\underline{\alpha}=(\alpha_1,\alpha_2,\ldots,\alpha_n)$, called the channel reliability vector, with the transmission of a codeword. These positive numbers give a measure of the reliability of the received symbols if the receiver were forced to make a hard decision. Thus if α_i is large, it is more probable that a decision made by the receiver upon the transmitted symbol is correct. We can use this channel reliability information to aid in selecting a subset of the parity check trellis to be searched. At the first k node levels for an (n,k) linear code each node on the trellis will have 2 branches emanating from it. The width of the trellis is bounded by $\min\{2^{n-k},2^k\}$.

As an example, the trellis for the Golay (24,12) code contains $2^{12}=4096$ nodes or states at its widest point. There is no advantage in using the VA for this code since word correlation decoding gives the same performance with about the same complexity.

If we receive a certain n-tuple $\underline{r}$ and several of the reliability numbers are large, we may accept the corresponding hard decision outputs as correct. In terms of the trellis this means that at certain node levels we extend the trellis along only the branch corresponding to the hard decision receiver output. At certain reliable node levels, then, the trellis does not grow in width, since there will be only one branch emanating from each node. The VA applied to a subtrellis generated in this way is equivalent to correlating the received vector $\underline{r}$ against a subset of the codewords. This set is generated by accepting the receiver hard decision outputs for the symbols in the k-p most likely information positions and permuting each of the remaining symbols. The size of the subset of code words, and hence the complexity of the corresponding subtrellis, will be a function of the number of digits permuted p.

The full algorithm proceeds by first generating a subtrellis by permuting p of the information symbols. This subtrellis is then searched up to the k'th node level, extending the M most likely paths at each node level. Finally, after the k'th node level, the VA is applied to extend the surviving paths to the final all zeroes node and select a codeword.

The parity check trellis for the Hamming (7,4) code is shown in Fig. 2. An example of a subtrellis generated for this code is shown in Fig. 3. In this case we have taken p=2. The real received vector $\underline{r}$ is shown along with the reliability vector $\underline{\alpha}$ and the corresponding receiver hard decision estimate.



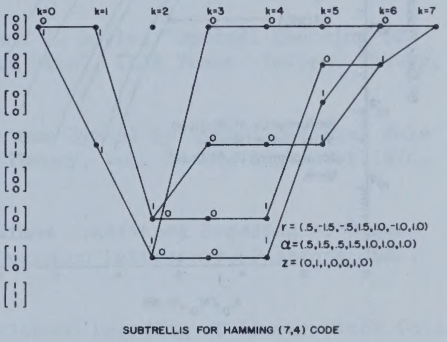

k=0
k=1
k=2
k=3
k=4
k=5
k=6
k=7
r = (.5,-1.5,-.5,1.5,1.0,-1.0,1.0)
α=(.5,1.5,.5,1.5,1.0,1.0,1.0)
z = (0,1,1,0,0,1,0)
SUBTRELLIS FOR HAMMING (7,4) CODE

Figure 2
Terminated Trellis for Hamming (7,4) code

Figure 3
Subtrellis for Hamming(7,4)code

IV. SIMULATION RESULTS

We have performed extensive simulations of the algorithm on the R.P.I. Interactive Communication Simulator (ICS) [7]. This is an extensive hardware-software system for interactive simulation of a variety of point-to-point communication systems. In addition to the decoder, the other components of the system model included a binary equiprobable source, a block encoder, a BPSK modulator and demodulator and an AWGN channel. To insure statistical accuracy over the range of signal to noise ratios which were simulated, 10^6 symbols were processed by the system for each test case.

The codes we chose for the simulations were the Hamming (7,4) and Golay (23,12) perfect codes along with their extended versions, the Hamming (8,4) and Golay (24,12). We discuss only the results for the extended codes. Results for the perfect codes are similar.

In the first set of simulations illustrated in Fig.'s 4 and 5, M was set equal to the maximum width of the trellis. In this case, no nodes are purged from the list at any node level. At the same time, p (the number of digits permuted) was varied and a simulation was performed for each choice of this parameter. For the Hamming code, we have chosen p=0,1,2,3,

4. The case p=0 corresponds to simply accepting the receiver hard decision outputs and thus the upper curve in the figure represents uncoded performance. For purposes of comparison we have also displayed the performance of a conventional algebraic decoder. Note that for p=2 the performance of the trellis decoder is about .5dB better than that of the conventional decoder, whereas p=3 gives performance nearly equal to maximum likelihood.

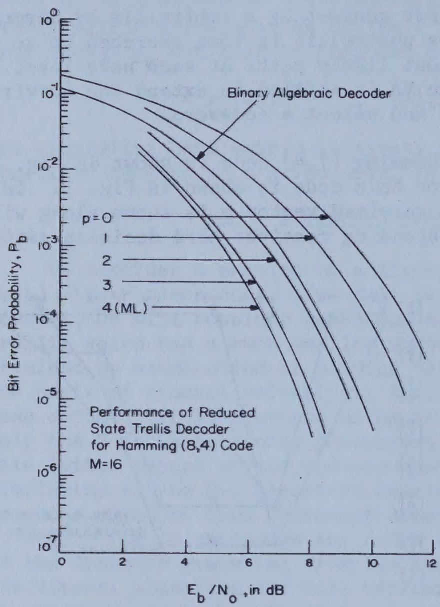


Figure 4

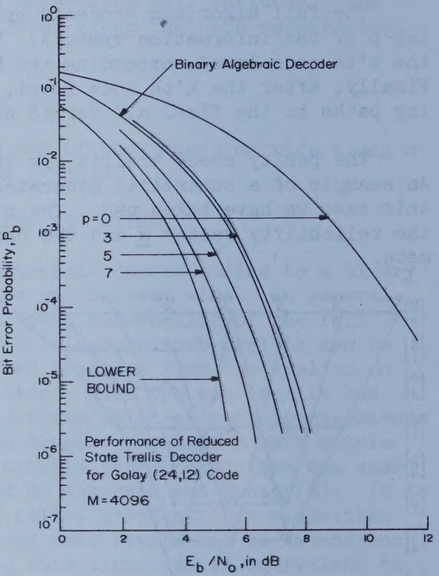


Figure 5

For the Golay code, we have chosen p=0,3,5,7. The full trellis for the Golay code was too complex to search within a reasonable time and therefore the maximum likelihood performance p=12 is not shown. We have included a lower bound on correlation decoding (cf. [1]) for purposes of comparison. Note that for the Golay code, the choice of p=3 gives performance nearly equal to the conventional decoder.

In the second set of simulations (Fig's. 6,7) we have set p to its maximum value and have purged all but M nodes at each node level. For the Hamming code, we have chosen M=1,2,4,8,16. For M=4 the trellis decoder performs about .8dB better than the conventional decoder and M=8 gives essentially maximum likelihood performance. For the Golary code, we have chosen M=1,4,16,64,128. In this case, M=16 gives performance comparable to the conventional decoder.

In the third and final set of simulations (Fig.'s 8,9) we have set the value of M to a fixed intermediate value and varied p. A simulation was again performed for each choice of p. For the Hamming code, we chose M=4 and p=1,2,3. For the case p=3, the performance is essentially the same as in Fig. 8 for M=4. For the Golay code, we chose M=64 and p=3,5,7,9. For p=7 the performance is about the same as in Fig. 7 for M=64.

V. SUMMARY AND CONCLUSIONS

We have described a general algorithm which performs a reduced-state search of an appropriately defined trellis. This algorithm has been applied to the decoding of several linear block codes and through computer simulation we have demonstrated the tradeoff between computational complexity and bit error performance. Interesting research directions would be to develop explicit performance bounds as a function of M and p and to apply the algorithm in the decoding of convolutional codes.

REFERENCES

1. D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement Information", IEEE Trans. Inform. Theory, Vol. IT-18, Jan. 1972.

2. L. D. Baumert and R. K. McElice, "Soft Decision Decoding of Block Codes", The Deep Space Progress Report 42-47, Jet Propulsion Laboratory, Pasadena, CA. August 1978, pp. 60-64.

3. L. R. Bahl, J. Cooke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", IEEE Trans. Inform. Theory, Vol. IT-20, March 1974, pp. 284-287.

4. C. Hartmann and L. Rudolph, "An Optimum Symbol By Symbol Decoding Rule for Linear Codes", IEEE Trans. Inform. Theory, Vol. It-22, September 1976, pp. 514-517.

5. H. J. Greenberger, "Approximate Maximum Likelihood Decoding of Block Codes", JPL Publication 78-107, Jet Propulsion Laboratory, Pasadena, CA., February 1978.

6. J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis", IEEE Trans. Inform. Theory, Vol. IT-24, January 1978.

7. J. Modestino, G. R. Redinbo, K. R. Matis and K. Y. Jung, "Digital Simulation of Communication Systems", Proc. of National Telecommunications Conf., 1979.

8. J. B. Anderson and S. Mohan, "A Systematic Analysis of Cost for Sequential Coding Algorithms", CRL Internal Report Series No. CRL-56, Communications Research Laboratory, McMaster University, Hamilton, Ontario, Canada.

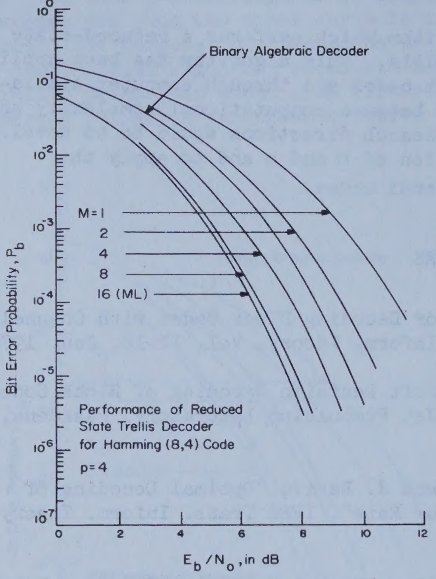


Figure 6

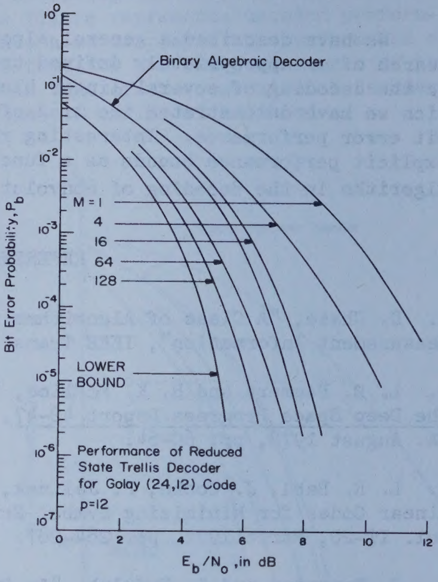


Figure 7

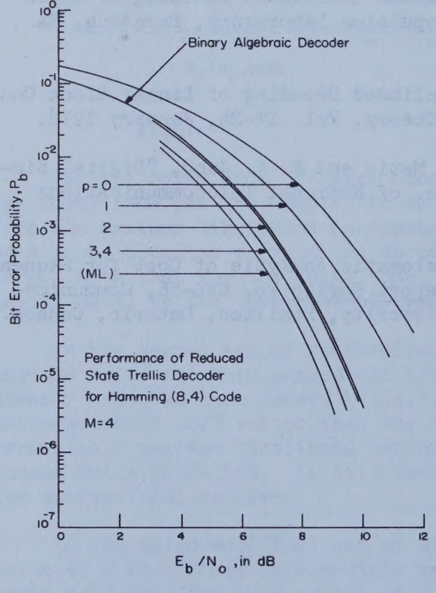


Figure 8

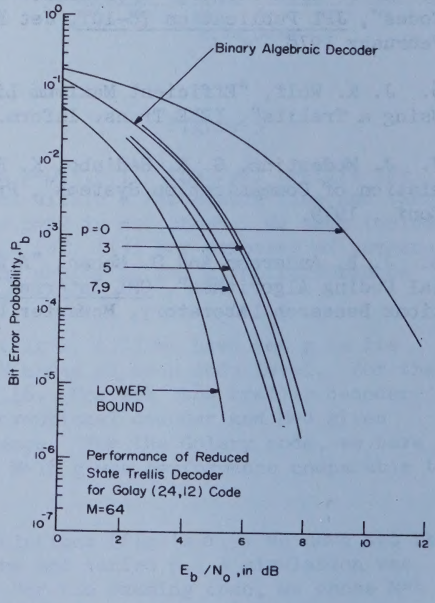


Figure 9

VORONOI REGIONS, SPHERE PACKINGS AND QUANTIZATION

N. J. A. SLOANE
Bell Laboratories
Murray Hill, New Jersey 07974

SUMMARY

A code for the Gaussian channel may be represented by a constellation of points in n-dimensional Euclidean space. The Voronoi region around a code word is the set of points that are closer to that codeword than to any other. In this paper the Voronoi regions are determined for a number of codes constructed from dense sphere packings. Full details will be published elsewhere.

INDEPENDENCE OF ERROR EVENT WAITING TIMES IN CONVOLUTIONAL DECODERS

K. Srivatsa & J.B. Anderson
Department of Electrical and Computer Engineering and Communications
Research Lab, McMaster University, Hamilton, Ontario L8S 4L7, Canada

ABSTRACT

As is well known, Viterbi decoding of convolutional codes gives the maximum likelihood estimate of an information sequence by an exhaustive search through the trellis. Bit errors occur during error events. In this paper we present statistics supporting the hypothesis that error events constitute a renewal process; i.e., the waiting times between error events are apparently independent. This behaviour is true for both rate-1/2 and rate-1/3 codes, even at low SNR and suboptimal decoder memory lengths. A test for their independence is carried out and a distribution is hypothesized. The independence greatly simplifies calculation of error statistics.

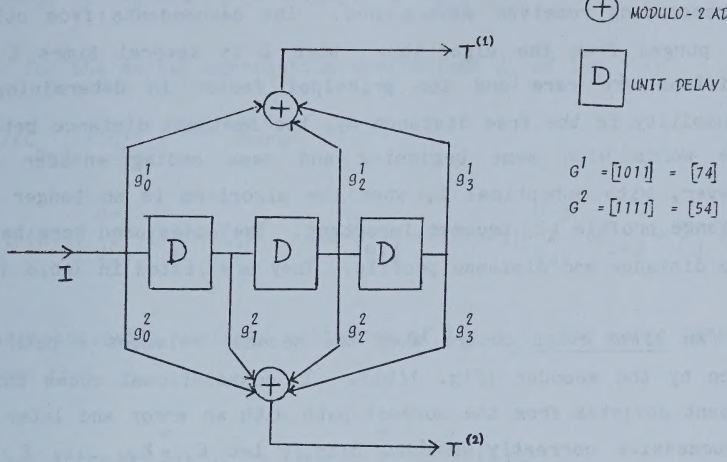
1. Codes and Decoding

A detailed discussion of convolutional codes and the Viterbi algorithm can be found in Viterbi and Omura [1]. Here we briefly introduce the necessary definitions. We have worked with codes of rate-1/2 and rate-1/3 reported by Johannesson [2] and Larsen [3] respectively. In a rate-1/2 code the information sequence I is convoluted with generators G^1 and G^2 to produce two output sequences $T^{(1)}$ and $T^{(2)}$. If M is the memory of the encoder (Fig. 1(a)) then

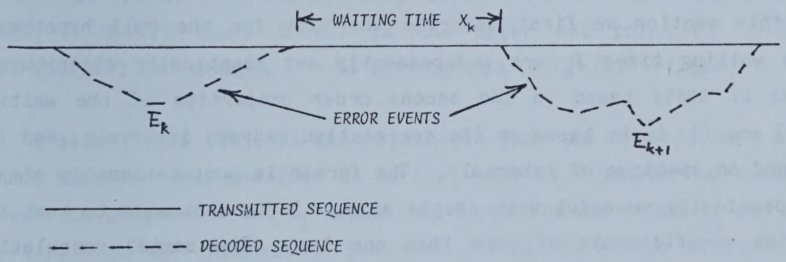
$$G^k = [g_o^k, g_1^k, \ldots\ldots, g_M^k], \; k = 1,2$$

The code length constraint length K equals M + 1. The generators are represented in the familiar octal notation where the first octal digit is given by $g_o^k \; g_1^k \; g_2^k$ etc., and similarly for the 3 generators of a rate-1/3 code.

In Viterbi decoding, the path memory length of the decision depth L is an important design parameter. The output symbol is released after a delay of L more observations from the corresponding received branch. Suppose the survivors entering each state in the trellis descend from more than one branch of the decision node. Then a decision is enforced in favour of the


MODULO-2 ADDER
D UNIT DELAY
$T^{(1)}$
g_0^1
g_2^1
g_3^1
$G^1 = [1011] = [74]$
$G^2 = [1111] = [54]$
I
D
D
D
g_0^2
g_1^2
g_2^2
g_3^2
$T^{(2)}$

(a) Rate-1/2 convolutional encoder


WAITING TIME X_k
E_k
ERROR EVENTS
E_{k+1}
TRANSMITTED SEQUENCE
DECODED SEQUENCE

(b) Error events and waiting times.

FIG. 1

Rate-1/2			Rate-1/3		
Generators	d_f	K	Generators	d_f	K
74 , 54	6	4	54 , 64,74	10	4
62 , 56	7	5	47 , 53,75	13	6
75 , 55	8	6			

TABLE 1. Rate-1/2 and rate-1/3 codes with their free distance d_f and constraint length K.

branch which is the first symbol of the descendant nearest to the corresponding received subsequence. The descendants from other branches are purged from the algorithm. When L is several times K such forced decisions are rare and the principal factor in determining the error probability is the free distance d_f, the smallest distance between any two code words with same beginning and same ending encoder states [1]. However, with suboptimal L, when the algorithm is no longer an MLD, the distance profile [2] becomes important. The codes used here have both good free distance and distance profile. They are listed in Table 1.

An error event occurs when the decoder releases a path segment not taken by the encoder (Fig. 1(b)). In convolutional codes this incorrect segment deviates from the correct path with an error and later merges with M successive correctly decoded bits. Let E_1, E_2, ..., E_n be n error events; then the waiting time between the end of E_k and the beginning of E_{k+1} inclusive will be denoted by X_k.

2. Independence of $\{X_k\}$ and a Distribution Hypothesis

In this section we first describe the test for the null hypothesis that the waiting times X_k are independently and identically distributed. Two types of tests based on the second order properties of the waiting times [4] are (1) tests based on the correlation between intervals, and (2) tests based on spectrum of intervals. The former is computationally simple and asymptotically powerful with sample size. It is desirable to test for correlation co-efficients of more than one lag. The serial correlation test [4] is described as follows.

Assuming that the series $\{X_k\}$ is stationary the estimate of the mean E[X] is given by

$$\overline{X} = (1/n) \sum_{i=1}^{n} X_i$$

In the absence of knowledge about E[X], the estimator for the serial covariances $C_j = E[\{X_i - E[X]\}\{X_{i+j} - E[X]\}]$ is given by

$$\tilde{C}_j = 1/(n-j) \sum_{i=1}^{n-j} (X_i - \overline{X_j'}) (X_{i+j} - \overline{X_j''})$$

where

$$\overline{X'_j} = 1/(n-j) \sum_{i=1}^{n-j} X_i \quad \text{and} \quad \overline{X''_j} = 1/(n-j) \sum_{i=1}^{n-j} X_{i+j}$$

The estimator for the serial correlation coefficient ρ_j of lag j is

$$\tilde{\rho}_j = \tilde{C}_j/(\tilde{C}_{o,j} \cdot \tilde{C}''_{o,j})^{1/2}, \text{ where}$$

$$\tilde{C}'_{o,j} = 1/(n-j) \sum_{i=1}^{n-j} (X_i - \overline{X'_j})^2 \quad \text{and} \quad \tilde{C}''_{o,j} = 1/(n-j) \sum_{i=1}^{n-j} (X_{i+j} - \overline{X''_j})^2$$

This is an asymptotically unbiased estimator of the ρ_j. For large samples it can be shown that

$$\mathrm{Var}\,(\tilde{\rho}_j) = 1/(n-j) \mathop{S}_{i=-\infty}^{\infty} \rho_i^2 = (1 + 2 \sum_{i=1}^{\infty} \rho_i^2)/(n-j)$$

For a renewal process $\rho_i = 0$, $i \neq 0$, so that $\mathrm{Var}\,(\tilde{\rho}_j) \sim 1/(n-j)$. Further, the asymptotic normality of $\tilde{\rho}_j$ can be proved for a variety of processes. Hence for a renewal process $\tilde{\rho}_j\sqrt{n-j}$ has a unit normal distribution for large n, so that the test at significance level α is to reject independence if $\tilde{\rho}_j\sqrt{n-j} > C(\alpha/2)$ where $C(\alpha/2)$ is the upper $\alpha/2$ point of unit normal distribution; for example $\alpha = 5\%$ corresponds to $C(\alpha/2) = 1.96$.

A second test concerns the hypothesis that the error events strike at random whenever they are free to strike, that is, whenever an event is not already in progress. Let l_i denote the length of the event E_i and let N be the length of the information sequence. Then the number of opportunities O_n for n events to strike is given by

$$O_n = N + n - \sum_{i=1}^{n} l_i$$

the event frequency is defined as $e_f = n/O_n$. Under the hypothesis, the waiting time X is geometrically distributed, so that Prob $\{X > k\} = (1 - p)^k$; p is the probability that an event strikes in any interval and k an integer. So the log "backward" cumulative distribution function of X decreases linearly with k. The value of p obtained from such an experimental plot is compared with e_f to test the hypothesis.

3. Simulation Results

The statistics presented here were obtained for the Viterbi algorithm with hard decisions. We have no reason to expect different results under soft decision; data for soft decision decoding of phase codes appear in [5]. The all zeroes information sequence was used in the simulation and the Gaussian noise generated by log-trig method. 6000-80000 bits were used, depending on signal to noise ratio E_b/N_o, L and constraint length K. The results were taken for E_b/N_o in the range 1-4 dB and L in the range 6-42, causing the decoded bit error rate to vary from 0.2 to 0.003. Energies outside this range yield too few error events. As L decreases the number of events and the average length of an event increase, the sensitivity increasing with K. On an average, half the bits of an event are correct. Table 2 summarizes the results for rate-1/2 codes for different SNRs. Table 3 shows the same for the rate-1/3 codes. The values $|\tilde{\rho}_1\sqrt{n-1}|$, $|\tilde{\rho}_2\sqrt{n-2}|$, $|\tilde{\rho}_3\sqrt{n-3}|$ are listed in these Tables. It can be seen that out of forty simulations only twice have any of these values exceeded the constant 1.96 corresponding to 5% rejection of the independence hypothesis. This is acceptable since this constant is obtained for a significance level of 5%. The Tables also show the closeness of p to e_f. Some log backward cdf plots are shown in Figs. 2, 3, and 4 for codes (74, 54), (75,55), and (47, 53, 75) respectively.

In hypothesizing the independence of waiting times, the error events are treated as point processes. This independence in turn implies that the number of error events in non-overlapping intervals are independent of each other. This absence of error propagation could come about in two ways, namely, 1) the error event length l_{i+1} is independent of the preceding waiting time X_i and/or 2) the waiting time X_i is independent of the preceding event length l_i. Intuitively only the former looks plausible and this has been verified by simulation. Figure 5 shows the log cdf plots of waiting times conditioned on short events and long events at $E_b/N_o = 1$ dB and L = 6. The difference in slopes indicates the dependence of waiting times on event lengths. When L is increased to 32 this difference becomes smaller as can be seen from Fig. 6. It has also been verified that, increasing SNR weakens the dependence for a given L.

Regarding the second test mentioned in 2, the agreement of the second and fourth columns in the Tables verifies the random arrival hypothesis.

E_b/N_0 = 4 dB, Rate = 1/2

L	p from log cdf	# events n	Event freq. e_f	$\lvert\tilde{\rho}_1\sqrt{n-1}\rvert$	$\lvert\tilde{\rho}_2\sqrt{n-2}\rvert$	$\lvert\tilde{\rho}_3\sqrt{n-3}\rvert$	Event length	Errors per event	±σ Interval	Decoded bit error rate
Code (74,54)										
6	0.0047	85	0.0045	0.400	0.698	0.444	14.8	47.2 %	10.8 %	0.0297
12	0.0020	37	0.0019	0.153	0.976	1.021	13.4	49.4	16.4	0.0123
18	0.0016	27	0.0014	0.670	0.425	0.508	10.6	45,8	19.2	0.0066
32	0.0015	27	0.0014	0.400	0.168	0.867	10.2	45.4	19.2	0.0063
Code (62,56)										
14	0.0019	37	0.0019	0.639	0.551	0.555	14.2	46.4 %	16.4 %	0.0122
24	0.0010	22	0.0011	0.045	1.668	0.222	10.7	42.5	21.3	0.005
34	0.0012	22	0.0011	0.341	1.148	0.143	10.7	42.5	21.3	0.005
Code (75,55)										
16	0.0012	24	0.0012	0.336	1.030	1.464	22.8	38.2 %	20.4 %	0.0105
24	0.0009	15	0.0008	1.706	0.198	0.682	17.7	40.3	25.8	0.0054
42	0.0007	11	0.0006	1.540	0.296	0.167	14.8	38.7	30.2	0.0032
E_b/N_0 = 1 dB, Rate = 1/2										
Code (74,54)										
6	0.0357	130	0.0364	1.610	0.525	0.807	18.7	49.7 %	8.7 %	0.2012
12	0.0244	110	0.0257	1.096	1.970	0.541	15.7	50.2	9.5	0.1445
18	0.0222	100	0.0231	0.684	1.116	0.393	16.7	49.5	10.0	0.1378
32	0.0212	93	0.0204	1.400	0.794	0.432	15.4	49.4	10.3	0.1180
Code (62,56)										
14	0.0250	90	0.0249	0.390	1.608	0.773	26.5	50.0 %	10.5 %	0.1987
24	0.0240	88	0.0235	0.261	0.505	0.428	25.6	49.9	10.6	0.1873
34	0.0216	83	0.0217	0.598	0.410	1.163	26.2	50.3	10.9	0.1822
44	0.0214	75	0.0195	0.168	0.152	0.969	28.6	50.4	11.5	0.1803
Code (75,55)										
16	0.0257	80	0.0230	1.104	1.180	0.672	31.5	42.6 %	11.2 %	0.179
24	0.0212	73	0.0201	0.608	1.353	0.472	32.5	42.3	11.7	0.1673
42	0.0198	70	0.0186	0.0856	1.842	0.607	31.9	42.3	12.0	0.1575
E_b/N_0 = 2 dB, Rate = 1/2										
Code (74,54)										
6	0.0199	293	0.0201	0.735	1.666	0.851	18.5	48.9 %	5.8 %	0.1327
12	0.0142	239	0.0141	0.043	0.660	0.338	12.9	48.8	6.5	0.0753
18	0.0123	212	0.0122	1.130	0.159	0.308	12.7	48.0	6.9	0.0648
32	0.0109	199	0.0114	0.295	1.464	0.190	11.7	47.9	7.1	0.0557
Code (62,56)										
14	0.0137	207	0.0131	0.081	0.922	0.009	20.3	49.1 %	6.9 %	0.1030
24	0.0117	180	0.0109	0.087	0.679	2.060 *	19.5	48.7	7.5	0.0855
34	0.1020	165	0.0107	0.385	1.893	0.281	17.9	48.0	7.8	0.0709
44	0.0097	164	0.0096	0.951	0.420	0.415	18.2	47.9	7.8	0.0715
Code (75,55)										
16	0.0117	172	0.0113	0.581	0.331	1.017	27.6	40.8 %	7.6 %	0.0968
24	0.0093	149	0.0092	0.752	0.160	0.435	26.0	40.2	8.1	0.0780
42	0.0081	132	0.0079	0.011	0.181	0.239	24.7	41.1	8.7	0.0670

TABLE 2, TABLE 3 (Overleaf) . Decoded error events at memory length L. Shown are prob. of an event p from log cdf plot, actual prob. e_f , the values obtained from correlation co-effecients ρ_1, ρ_2, ρ_3 and number of events n ; one σ confidence intervals are derived from n . '*' indicates rejection of independance of waiting times.

E_b/N_0 = 2.8 dB, Rate = 1/3

L	p from log cdf	# events n	Event freq. e_f	$\lvert\tilde{F}_1\sqrt{n-1}\rvert$	$\lvert\tilde{F}_2\sqrt{n-2}\rvert$	$\lvert\tilde{F}_3\sqrt{n-3}\rvert$	Event length	Errors per event	±σ Interval	Decoded bit error rate
Code (54,64,74)										
4	0.0134	218	0.0139	0.779	0.772	0.708	19.9	48.2 %	6.7 %	0.1043
24	0.0035	66	0.0034	0.467	0.935	0.075	12.1	43.5	12.3	0.0174
Code (47,53,75)										
4	0.0118	156	0.0123	1.545	1.081	0.309	47.2	48.9 %	8.0 %	0.1798
24	0.0035	68	0.0037	0.989	1.054	1.255	21.3	44.6	12.1	0.0328
E /N = 3.8 dB, Rate = 1/3										
Code (54,64,74)										
4	0.0064	227	0.0063	0.628	0.072	0.053	16.4	48.2 %	6.6 %	0.0448
24	0.0009	66	0.0008	0.279	0.139	1.263	10.0	41.5	12.3	0.0034
Code (47,53,75)										
6	0.0050	87	0.0051	0.007	1.270	2.180 *	31.7	50.5 %	10.7 %	0.0697
24	0.0008	61	0.0008	1.143	0.429	0.746	15.2	40.3	12.8	0.0047

TABLE 3.

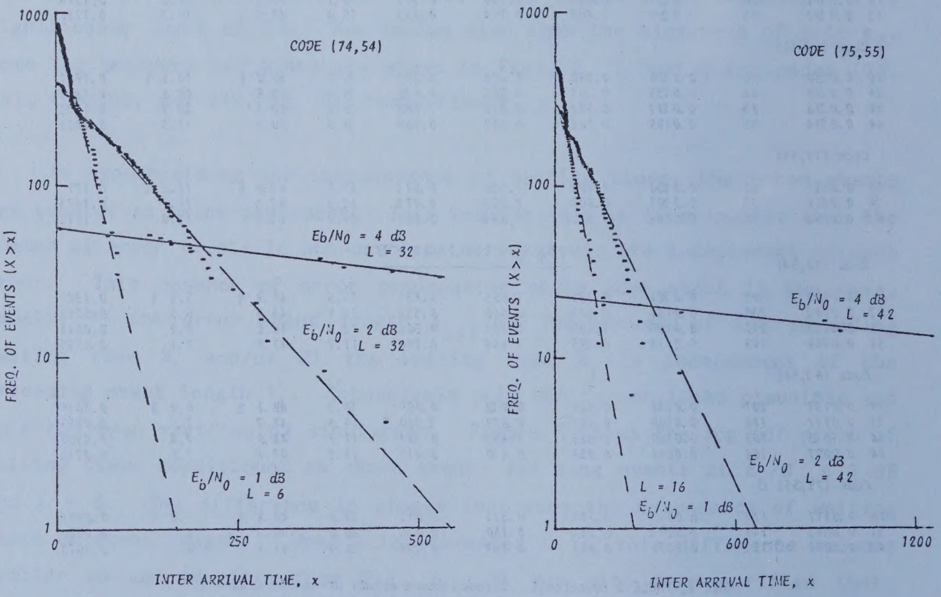


FIG.2 ,3 ,4. Log-scale "backward" cdf plots for waiting times .Event statistics are in Tables 2, 3.

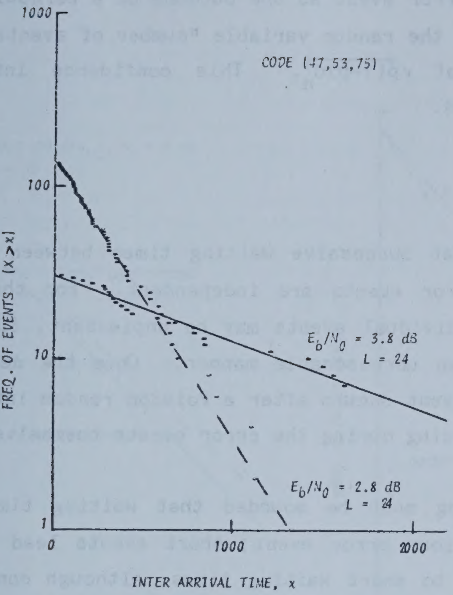


FIG. 4

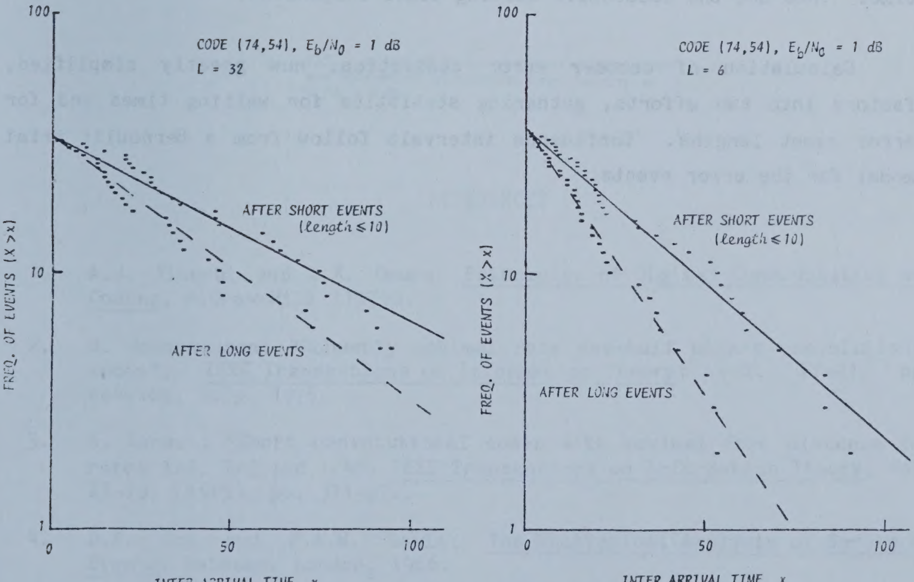


FIG. 5,6. Log cdf plots show the dependance of waiting times on previous event length for two different decision depths.

Agreement is relative to statistical confidence intervals. If we regard the occurrence of an error event as the outcome of a bernoulli trial with a success probability p, the random variable "number of events" has one sigma confidence interval of $\sqrt{p(1-p)o_n}$. This confidence interval is also indicated in the Tables.

4. Conclusions

We have shown that successive waiting times between Viterbi-decoded convolutional code error events are independent. For the decoder, this implies that while individual events may be unpleasant, successive events do not propagate in an unreasonable manner. Once the decoder enters an error event the next event occurs after a Poisson random interval, if we do not count time as elapsing during the error events themselves.

A word of warning must be sounded that waiting times are in fact dependent on the previous error event; short events lead to long waiting times and long events to short waiting times, although conditioned on the error event length, the waiting time is still apparently Poisson. Error event lengths, however, evidently do not depend on the previous waiting time. Thus are the successive waiting times indpendent.

Calculation of decoder error statistics, now greatly simplified, factors into two efforts, gathering statistics for waiting times and for error event lengths. Confidence intervals follow from a Bernoulli trial model for the error events.

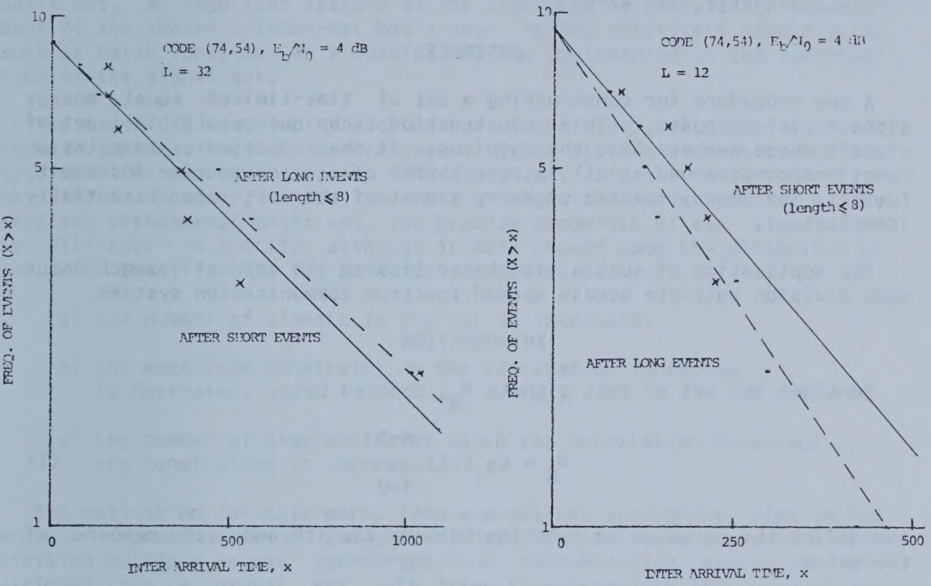


FIG. 7,8. Log cdf plots show that waiting times are independent of previous event lengths at E_b/N_0 = 4 dB.

REFERENCES

1. A.J. Viterbi and J.K. Omura; Principles of Digital Communication and Coding, McGraw-Hill (1979).

2. R. Johannesson; "Robustly optimal rate one-half binary convolutional codes", IEEE Transactions on Information Theory; vol. IT-21, pp. 464-468, July, 1975.

3. K. Larsen; "Short convolutional codes with maximal free distance for rates 1/2, 1/3 and 1/4", IEEE Transactions on Information Theory, vol. IT-19, (1973), pp. 371-372.

4. D.R. Cox and P.A.W. Lewis; The Statistical Analysis of Series of Events, Methuen, London, 1966.

5. J.B. Anderson; "Simulated error performance of multi-h phase codes", IEEE Trans. on Information Theory, 1980, vol. IT-27, May 1981.

DESIGN OF SIGNAL SETS WITH SPECIFIED CORRELATION PROPERTIES

THOMAS P. MCGREE
GEORGE R. COOPER
School of Electrical Engineering
Purdue University
W. Lafayette, IN 47907

ABSTRACT

A new procedure for constructing a set of time-limited equal energy signals is proposed. This construction technique results in a set of signals whose members have the magnitudes of their aperiodic correlation functions constrained at all integer shifts of a selected time increment. Further, the energy spectra of every signal of the set are essentially identical.

The application of such a signal set lies in the area of asynchronous code division multiple access spread spectrum communication systems.

INTRODUCTION

Consider the set of real signals Ω_M, denoted by

$$\Omega_M = \{s_i(t)\}_{i=0}^{M-1} \tag{1}$$

and define the correlation function between the ith and jth members of the set as

$$\psi_{ij}(\tau) = \int_0^T s_i(t)s_j(t-\tau)dt, \ |\tau|\leq T \tag{2}$$

The investigation of this paper deals with aperiodic correlation properties; hence, it is assumed that each member of Ω_M in (1) above exists only over the time interval [0,T].

Assuming all members of Ω_M have unit energy, i.e.,

$$\int_0^T s_i^2(t)dt = 1, \ i=0,1,\ldots,M-1 \tag{3}$$

desirable design criteria in terms of the correlation function can be stated as follows:

$$|\psi_{ii}(\tau)|\leq \alpha, \ i=0,1,\ldots M-1 \quad |\tau|\leq T, \ \tau\neq 0 \tag{4}$$

$$|\psi_{ij}(\tau)|\leq \beta, \ i\neq j \quad |\tau|\leq T \ \ i,j=0,1,\ldots,M-1 \tag{5}$$

Uniformly bounding the magnitude of the aperiodic correlation functions generated from any large set of signals is indeed a difficult task. In this paper, a design procedure, which is algorithmic in nature, is proposed that creates an arbitrarily sized signal set whose members have the magnitudes of their aperiodic correlation functions constrained at all integer shifts of a selected time increment.

The design procedure relies upon the properties of pseudonoise (PN) sequences [1] and the concept of signal expansion via an orthonormal basis set. An important feature of the algorithm is that it is independent of the chosen orthonormal basis set. Hence, relatively simple sets, such as Walsh functions or sinusoids, may be implemented in the construction of the signal set.

The bandwidth occupied by each member of the signal set is of interest in most applications. In general, each signal of this set has essentially identical energy spectra, and thus all signals have the same bandwidth. However, since the design algorithm is independent of the selected orthonormal basis set, the precise bandwidth of the signal set is difficult to specify, although it does depend upon the parameters of the set. Specifically, the bandwidth increases as

a) the number of signals in the set is increased,

b) the magnitude constraint on the correlation functions is decreased, and

c) the number of time shifts at which the correlation functions are constrained is increased.

The motivation for this work, from a practical standpoint, lies in the area of spread-spectrum communications. In particular, use in a code division multiple access spread-spectrum communication system is envisioned for a signal set with fixed time-bandwidth (TB) product and bounded correlation properties.

In such a system (see, for example, [2]), a number of different users share the same portion of the frequency spectrum simultaneously. Each user's receiver is usually a matched filter or the equivalent correlator structure. Hence, the correlation peaks of the signal set employed determine the interference that each user sees at his receiver due to all other users on the channel at the same time. This mutual interference limits the number of allowable simultaneous users permissible while still being able to maintain adequate system performance. [3]

NOTATIONAL CONVENTIONS FOR REPRESENTATION OF TIME DOMAIN SIGNALS

The signal set design procedure to be proposed here relies on representing a signal via its coefficients of a finite dimensional orthonormal basis set signal expansion. As will become evident, the design procedure forces the signals to be members of the finite dimensional space spanned by the chosen basis set. Therefore, there is no need to be concerned with signal approximation error, the representations are exact.

To begin the mathematical development, start with an arbitrary real signal, s(t) and define a time-limiting operator as:

$D_\ell \triangleq$ time-limiting an arbitrary time function, s(t), to an interval of length τ seconds centered at $t = \tau(\ell + 1/2)$

Therefore,

$$D_\ell[s(t)] = \begin{cases} s(t) & \ell\tau < t < (\ell+1)\tau \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

For the original s(t), let f(t) denote s(t) time-limited to the interval $\underline{0<t<T}$. Further, select T and τ such that $T = J\tau$, where J is an integer. Thus, f(t) can be specified as follows:

$$f(t) = \sum_{\ell=0}^{J-1} D_\ell[s(t)] \qquad (7)$$

Now, consider any set of real orthonormal basis functions complete in the time interval $[-\tau/2,\tau/2]$. Denote this set of basis functions by $\{\phi_n(t)\}$, where n ranges from 0 to infinity.

Hence, each time chip of s(t), represented by $D_\ell[s(t)]$, where ℓ indicates the ℓth time chip, can be approximated by an N+1 term signal expansion:

$$D_\ell[s(t)] \simeq \sum_{n=0}^{N} c_{n_\ell} \; D_\ell \; \{\phi_n[t - (\frac{2\ell+1}{2})\tau]\} \qquad (8)$$

So that

$$f(t) \simeq \sum_{\ell=0}^{J-1} \; \sum_{n=0}^{N} c_{n_\ell} D_\ell\{\phi_n[t - (\frac{2\ell+1}{2})\tau]\} \qquad (9)$$

where the c_{n_ℓ} are calculated from:

$$\int_{-\infty}^{\infty} f(t) D_k\{\phi_m[t - (\frac{2k+1}{2})\tau]\}dt = c_{m_k} \qquad (10)$$

Therefore,

$$c_{n_\ell} = \int_{\ell\tau}^{(\ell+1)\tau} f(t)\phi_n[t - (\frac{2\ell+1}{2})\tau]dt \qquad (11)$$

Note that in deriving equation (11), use was made of both the time-limiting property of the operator D and the orthonormality of the basis set over the time interval $(-\tau/2,\tau/2)$.

APERIODIC CORRELATION FUNCTIONS IN TERMS OF SIGNAL EXPANSION COEFFICIENTS

At this point, it is desired to express the autocorrelation function, crosscorrelation function and signal energy in terms of the signal expansion coefficients, c_{n_ℓ}.

First, the signal energy is obtained from:

$$\int_0^T f^2(t)dt = \int_0^T \left\{ \sum_{\ell=0}^{J-1} \; \sum_{n=0}^{N} c_{n_\ell} \; D_\ell\{\phi_n[t - (\frac{2\ell+1}{2})\tau]\}\right\}^2 dt \qquad (12)$$

Again, employing the time limiting of D and the orthonormality of the ϕ_n's, one obtains

$$\int_0^T f^2(t)dt = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N} c_{n_\ell}^2 \tag{13}$$

A derivation of the auto- or crosscorrelation function is not as straightforward as the signal energy expression. Hence, we are content to consider only time shifts of the form

$$\Delta t = p\tau \tag{14}$$

where p is an integer such that $-(J-1) \leq p \leq (J-1)$.

The autocorrelation function is now derived for $0 \leq p \leq J-1$; that is, for positive integral time shifts of τ. The derivation for negative p is exactly analogous to that of positive p, thus the result is merely stated. Keeping in mind that $p \geq 0$, the autocorrelation of f(t) is

$$\Psi_{ff}(p\tau) = \int_0^T f(t)f(t - p\tau)dt$$

$$= \int_0^T \sum_{\ell=0}^{J-1} \sum_{k=0}^{J-1} \sum_{n=0}^{N} \sum_{m=0}^{N} c_{n_\ell} c_{m_k} D_\ell\{\phi_n[t - (\frac{2\ell+1}{2})\tau]\} \cdot$$

$$D_{k+p}\{\phi_m[t - (\frac{2p+2k+1}{2})\tau]\}dt \tag{15}$$

In equation (15), overlap of the $D_\ell[\cdot]$ and $D_{k+p}[\cdot]$ terms will only occur when $(2\ell+1)$ equals $(2p + 2k + 1)$; or, equivalently, when ℓ equals p+k. Setting ℓ equal to p+k in equation (15) leads to

$$\int_0^T f(t)f(t-p\tau)dt = \sum_{k=0}^{J-1-p} \sum_{m=0}^{N} c_{m_{k+p}} c_{m_k} \ ; \ 0 \leq p \leq J-1 \tag{16}$$

As stated earlier, the derivation for negative p is analogous to the above, with the result being

$$\int_0^T f(t)f(t-p\tau)dt = \sum_{k=-p}^{J-1} \sum_{m=0}^{N} c_{m_{k+p}} c_{m_k} \ , \ -(J-1) \leq p \leq 0 \tag{17}$$

Therefore, combining equations (16) and (17), one obtains the general result

$$\int_0^T f(t)f(t-p\tau)dt = \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \sum_{m=0}^{N} c_{m_{k+p}} c_{m_k} \ , \ -(J-1) \leq p \leq (J-1) \tag{18}$$

The final step is to obtain an expression similar to equation (18) for the crosscorrelation function between two signals represented via this time chip signal expansion method. With f(t) as in equation (9), let

$$g(t) = \sum_{k=0}^{J-1} \sum_{m=0}^{N} b_{m_k} D_k\{\phi_m[t - (\frac{2k+1}{2})\tau]\} \qquad (19)$$

Proceeding in the same manner as followed in computing the autocorrelation function, the crosscorrelation becomes

$$\int_0^T f(t)g(t-p\tau)dt = \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \sum_{m=0}^{N} c_{m_{k+p}} b_{m_k}, \quad -(J-1) \leq p \leq (J-1) \qquad (20)$$

Summarizing the results to this point, equations (13), (18) and (20) yield expressions for signal energy, signal autocorrelation, and the crosscorrelation of two signals, respectively, in terms of the signal expansion coefficients from each of the J time chips.

SIGNAL SET DESIGN ALGORITHM

The groundwork has now been laid for the presentation of the signal set construction procedure. Assume that one is given the following requirements:

1) Require M signals in the set, $\{s_i(t)\}_{i=0}^{M-1}$.

2) Require each signal to be time-limited to $0 \leq t \leq T$.

3) Require $\int_0^T s_i^2(t)dt = 1$, for all i. This constraint is equivalent to any equi-finite energy constraint one wishes to impose.

4) With T divided into strips of width τ, where $J\tau = T$, J an integer, require

$$|\Psi_{ij}(p\tau)| = |\int_0^T s_i(t)s_j(t-p\tau)dt| \leq \alpha, \text{ where}$$

p is an integer such that $-(J-1) \leq p \leq J-1$.

With these constraints, the algorithm proceeds as follows:

1) Choose N, the number of orthonormal basis functions, such that

$$N = 2^L - 1 \quad , \quad L \text{ an integer}$$

and

$$N \geq \max [JM ; 1/\alpha]$$

2) Specify a pseudonoise (PN) sequence of period N. Denote the selected sequence by

$$\{a_0, a_1, \ldots, a_{N-1}\}, \text{ where } |a_i| = \frac{1}{\sqrt{NJ}}.$$

3) Select an orthonormal basis set $\{\phi_n(t)\}_{n=0}^{N-1}$, orthonormal over the time interval $[0,\tau]$.

4) Define each signal $s_k(t)$ of the set $\{s_i(t)\}_{i=0}^{M-1}$ as follows:

$$s_k(t) = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} a_{n \oplus \ell \oplus kJ} D_\ell\{\phi_n(t-\ell\tau)\},$$

where $\oplus$ denotes addition modulo N.

At this point a brief note on PN sequences is in order. The only property of these sequences that is of interest here is the nature of their periodic autocorrelation function, given by

$$\rho(i) = \sum_{j=0}^{N-1} a_j \cdot a_{i \oplus j} \ , \quad 0 \leq i \leq N-1 \tag{21}$$

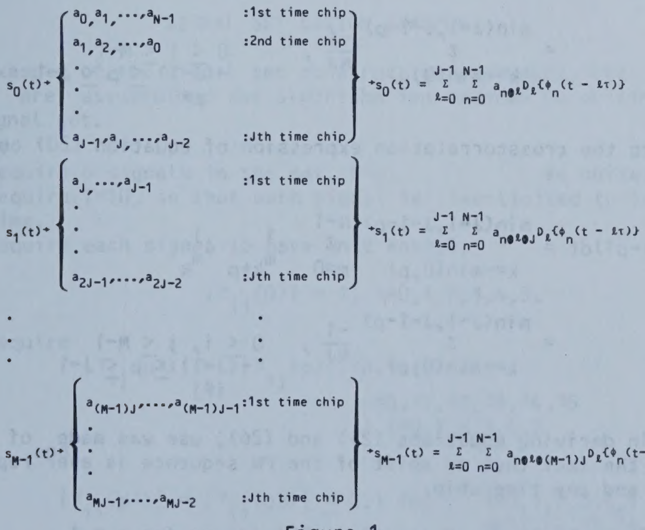
With the magnitude of the a_i's as specified in step 2 of the algorithm, $\rho(i)$ takes on the values

$$\rho(i) = \begin{cases} \dfrac{1}{J} \ , & i=0 \\ \dfrac{-1}{NJ} \ , & i \neq 0 \end{cases} \tag{22}$$

The magnitude of $\Psi_{ij}(p\tau)$ relies on this feature of PN sequences as described below.

It is also noted, as a matter of realization, that a PN sequence exists for all N where $N = 2^L - 1$, L an integer [1]. Further, the generation of PN sequences is accomplished with a simple feedback shift register configuration, defined by a primitive polynomial of appropriate degree. The interested reader is referred to [1].

Figure 1 is used to clarify the investigation of the properties of the resulting signal set specified by the above algorithm. Referring to Figure 1, note that the signal expansion coefficients for each signal for each time chip are determined by a periodic shifted version of the PN sequence. Each time chip requires N coefficients. Finally, it is noted that step 1 of the algorithm, the specification of N, insures that the greatest shift of the PN sequence, occurring for the Jth time chip of the M-1 signal, does not take the PN sequence through a whole cycle of shifts.



$s_0(t) \rightarrow$
$a_0, a_1, \cdots, a_{N-1}$:1st time chip
$a_1, a_2, \cdots, a_0$:2nd time chip
$a_{J-1}, a_J, \cdots, a_{J-2}$:Jth time chip
$\rightarrow s_0(t) = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} a_{n \oplus \ell} D_\ell \{\phi_n(t - \ell\tau)\}$
$s_1(t) \rightarrow$
$a_J, \cdots, a_{J-1}$:1st time chip
$a_{2J-1}, \cdots, a_{2J-2}$:Jth time chip
$\rightarrow s_1(t) = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} a_{n \oplus \ell \oplus J} D_\ell \{\phi_n(t - \ell\tau)\}$
$s_{M-1}(t) \rightarrow$
$a_{(M-1)J}, \cdots, a_{(M-1)J-1}$:1st time chip
$a_{MJ-1}, \cdots, a_{MJ-2}$:Jth time chip
$\rightarrow s_{M-1}(t) = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} a_{n \oplus \ell \oplus (M-1)J} D_\ell \{\phi_n(t-\ell\tau)\}$

Figure 1

Returning to the notation of equations (13), (18) and (20), let $\{c^i_{n_\ell}\}$ denote the signal expansion coefficients for the ith signal of the set. Referring to Figure 1, obtain

$$c^i_{n_\ell} = a_{n \oplus \ell \oplus iJ} \tag{23}$$

where $0 \le n \le N-1$, $0 \le \ell \le J-1$, and $0 \le i \le M-1$. To stress the point further, note that for ℓ and i fixed, $\{c^i_{n_\ell}\}$ for $0 \le n \le N-1$ is simply the $iJ+\ell$ shifted version of the PN sequence.

Hence, returning to the signal energy expression of equation (13), obtain the energy as

$$\begin{aligned}\int_0^T s_i^2(t)dt &= \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} (c^i_{n_\ell})^2 \\ &= \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} (a_{n \oplus \ell \oplus iJ})^2 \\ &= \sum_{\ell=0}^{J-1} \frac{1}{J} \\ &= 1 \text{ , } 0 \le i \le M-1 \end{aligned} \tag{24}$$

Returning to the autocorrelation expression of equation (18) obtain the autocorrelation as

$$\begin{aligned}\int_0^T s_i(t)s_i(t-p\tau)dt &= \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \sum_{m=0}^{N-1} c^i_{m_{k+p}} c^i_{m_k} \\ &= \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \frac{-1}{NJ}, \quad \begin{matrix} 0 \le i \le M-1 \\ -(J-1) \le p \le J-1 \\ p \ne 0 \end{matrix}\end{aligned} \tag{25}$$

Returning to the crosscorrelation expression of equation (20) obtain the crosscorrelation as

$$\begin{aligned}\int_0^T s_i(t)s_j(t-p\tau)dt &= \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \sum_{m=0}^{N-1} c^i_{m_{k+p}} c^j_{m_k} \\ &= \sum_{k=-\min(0,p)}^{\min(J-1,J-1-p)} \frac{-1}{NJ}, \quad \begin{matrix} 0 \le i, j \le M-1 \\ -(J-1) \le p \le J-1 \\ i \ne j \end{matrix}\end{aligned} \tag{26}$$

Note that in deriving equations (25) and (26), use was made of equation (22) and the fact that no shift of the PN sequence is ever repeated for any signal and any time chip.

The maximum magnitude of equation (25) occurs for p = 1,

$$|\Psi_{ii}(\tau)| = \frac{J-1}{NJ} < \frac{1}{N} \tag{27}$$

while the maximum magnitude of equation (26) occurs for p = 0,

$$|\Psi_{ij}(0)| = \frac{1}{N} \tag{28}$$

Returning to step 1 of the algorithm, it is clear that N has been specified such that

$$\frac{1}{N} \leq \alpha \tag{29}$$

Therefore, equations (27) and (28) indicate that the cross and autocorrelation functions for all integer shifts of τ have been successfully bounded for all M signals of the designed set.

Finally, examination of step 4 of the signal design algorithm justifies the claim given in the introduction that all signals of the set have the same bandwidth.

A signal's energy spectra is defined as the squared magnitude of its Fourier Transform. Hence, the signal defining equation of step 4 forces every signal of the set to have essentially identical energy spectra. This is because every time chip of each signal is spanned by the same orthonormal basis set, with the signal expansion coefficients from signal to signal merely being a periodic rotation of the same PN sequence, as indicated by equation (23).

Step 1 of the algorithm indicates why, in general, changing the signal set parameters in a manner described by points a, b or c of the introduction increases the bandwidth of the set. This is simply because any of these changes tend to increase the number of basis functions required to span each time chip, as specified by the inequality on N of step 1. It is a well known fact that increasing the number of orthonormal functions occupying the same time interval forces the bandwidth of the orthonormal set to increase. Thus, the bandwidth of the signal set is likewise increased.

SIGNAL SET DESIGN EXAMPLE

As an example of the signal set construction procedure, the following parameters are assumed and the algorithm implemented to obtain a satisfactory signal set.

Parameters:

1) Require 6 signals in the set, M=6.
2) Require T=10, so that each signal is time-limited to 10 units of time.
3) Require each signal to have unit energy,

$$|\Psi_{ii}(0)| = 1, \; i=0,1,2,3,4,5.$$

4) Require

$$|\Psi_{ij}(p\tau)| = |\Psi_{ij}(p2)| \leq 0.1 \text{ for } i \neq j \quad \begin{matrix} p=0,\pm1,\pm2,\pm3,\pm4,\pm5 \\ i,j=0,1,2,3,4,5 \end{matrix}$$

and

$$|\Psi_{ii}(p\tau)| = |\Psi_{ii}(p2)| \leq 0.1 \text{ for } \quad \begin{matrix} i=0,1,2,3,4,5 \\ p=\pm1,\pm2,\pm3,\pm4,\pm5 \end{matrix}$$

Hence, for the notation used previously, $\tau = 2$, $\alpha = 0.1$ and $J = 5$.

The algorithm proceeds as follows:

1) Select orthonormal sinusoids for the basis set. Hence,

$$\phi_n(t) = \begin{cases} \sqrt{\frac{2}{\tau}}\sin\left[\frac{2\pi(n+1)t}{\tau}\right] = \sin[\pi(n+1)t]\,, & 0 \leq t \leq 2 \\ 0 & , \text{elsewhere} \end{cases}$$

2) Calculation of N, length of PN sequence and number of basis functions to be employed:

$$N_1 \geq JM = 5 \cdot 6 = 30$$

$$N_1 \geq 1/\alpha = 10 \qquad (\alpha = 0.1)$$

Hence, $N = 2^5 - 1 = 31$ will suffice.

3) Generation of the PN sequence:

The shift register specified by the primitive polynomial $h(x) = x^5 + x^2 + 1$ will generate the required sequence (see [1]). Denote the resulting sequence by $\{a_0, a_1, \ldots, a_{30}\}$, where

$$a_i = \frac{\pm 1}{\sqrt{NJ}} = \frac{\pm 1}{\sqrt{31 \cdot 5}} = \frac{\pm 1}{\sqrt{155}}$$

4) Thus,

$$s_k(t) = \sum_{\ell=0}^{J-1} \sum_{n=0}^{N-1} a_{n \oplus \ell \oplus kJ} D_\ell\{\sin[\pi(n+1)(t - \ell\tau)]\}$$

$$= \sum_{\ell=0}^{4} \sum_{n=0}^{30} a_{n \oplus \ell \oplus k5} D_\ell\{\sin[\pi(n+1)(t - 2\ell)]\}$$

for $k = 0, 1, 2, 3, 4, 5$ and where D_ℓ is an operator that time limits a function to

$\ell\tau < t < (\ell+1)\tau \to 2\ell < t < 2(\ell+1)$.

Correlation functions generated by signals coming from the above set have been computed numerically. The magnitude of a representative autocorrelation function for positive time shifts is plotted as Figure 2, while Figure 3 illustrates a typical crosscorrelation function.

Note that in addition to the magnitude of these functions being bounded by 0.1 at integer shifts of 2, they are nearly uniformly bounded. Indeed, these correlation functions contain few peaks greater than 0.1, with all such peaks being less than 0.2.

REFERENCES

1.) MacWilliams, F. J. and Sloane, N. J. A., "Pseudo-Random Sequences and Arrays," Proceedings of the IEEE, vol. 64, no. 12, pp 1715-1729, December 1976.

2.) Cooper, G. R. and Nettleton, R. W., "A Spread-Spectrum Technique for High-Capacity Mobile Communications," IEEE Trans. on Vehicular Technology, vol. VT-27, no. 4, pp. 264-275, November 1978.

3.) Rowe, H. E., "Bounds on Spread-Spectrum Systems," 1979 Midcon Professional Program, Session 15, November 6-8, 1979.

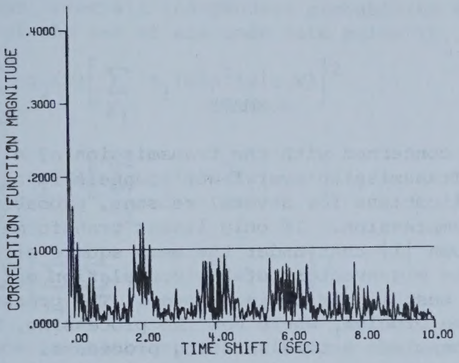


Figure 2: Typical Autocorrelation Function

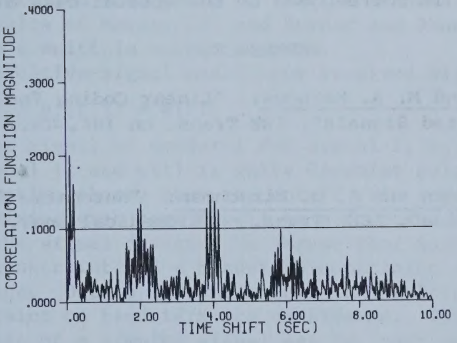


Figure 3: Typical Crosscorrelation Function

EFFICIENT TRANSMISSION OF A SET OF SIGNALS OVER FEWER CHANNELS BY LINEAR TRANSFORMATIONS AND TIME SHIFTING

N. KALOUPTSIDIS, C. CAROUBALOS, D. MANOLAKIS
Department of Electronics
University of Athens
Athens, Greece

SUMMARY

This paper is concerned with the transmission of n signals over m channels, $m \leq n$. Transmission over fewer channels is of great interest in many practical applications for several reasons, probably the most significant being, data compression. If only linear transformations are performed, then it is well known |1| that under the mean square error criterion, the matrix formed by the eigenvectors of the correlation matrix of the original signals yields the best linear coding scheme. The present paper is concerned with the same problem, where now the processing, besides linear transformations, comprises a time shifting procedure. Consideration of time shifting to a great extent was motivated by the behaviour that several signals exhibit (solar radio bursts in particular, or even ECG and geophysical signals), as being almost shifts of one another. Necessary conditions for the optimum time shifts as well as the linear coder are derived. It turns out that the resulting equations are coupled and highly nonlinear, thus various iterative schemes are developed to provide for the optimum solution. Finally it is indicated that a similar approach pursued in |2| although stated differently, it corresponds to the special case m = 1 of this work.

REFERENCES

1. H. P. Kramer and M. A. Mathews: "Linear Coding for Transmitting a Set of Correlated Signals", IRE Trans. on Inf. Th., Vol. 2, pp. 41-46, Sept. 1956.

2. R. A. Christensen and A. D. Hirschman: "Automatic Phase Alignment for the k-L Expansion", IEEE Trans. on Biomedical Eng., Vol. 26, No. 2, Feb. 1979.

MODULATOR DESIGN FOR A BANDLIMITED MULTIPLE ACCESS CHANNEL*

PRAKASH NARAYAN and DONALD L. SNYDER
Washington University
St. Louis, Missouri 63130

We summarize our results for a multiple access channel having two senders and one receiver. The extension to n senders is straightforward, and the results are similar. The two sender cutoff rate region is the convex hull of the union, over all independent probability distributions $q(u,v)=q_1(u)q_2(v)$, of the set of all code rate pairs (R_1,R_2) that satisfy

$$R_1 \leq -\log \sum_{Y} \sum_{X_2} q_2(v)\left[\sum_{X_1} q_1(u)p^{\frac{1}{2}}(w|u,v)\right]^2,$$

$$R_2 \leq -\log \sum_{Y} \sum_{X_2} q_1(u)\left[\sum_{X_1} q_2(v)p^{\frac{1}{2}}(w|u,v)\right]^2, \qquad (1)$$

and

$$R_1 + R_2 \leq -\log \sum_{Y}\left[\sum_{X_1} \sum_{X_2} q(u,v)p^{\frac{1}{2}}(w|u,v)\right]^2,$$

where $p(w|u,v)$ is the transistion probability of the channel, w is a letter in the output alphabet Y, u is a letter in the code alphabet X_1 of sender-1, and v is a letter in the code alphabet X_2 of sender-2. This cutoff rate region generalizes the cutoff rate parameter [1] of a single user channel to a multiple access channel. Bounds on the average decoding-error probability for both block and convolutional coding can be given in terms of the cutoff rate region for the multiple access channel. Our effort in this research is to use the cutoff rate region to study the coordinated design of the encoders and modulators of the two senders in a manner that generalizes the results of Massey [1] and Snyder and Rhodes [2] for a single user channel to a multiple access channel.

For an AWGN, additive-signal model, the received signal is

$$r(t) = s_1(t;i) + s_2(t;j) + w(t),$$

where $s_1(t;i)$ is the signal of sender-1 for symbol i, $s_2(t;j)$ is the signal of sender-2 for symbol j, and $w(t)$ is white Gaussian noise, we find using constrained optimization theory that the cutoff rate region is maximized by independently maximizing each term in (1) as a function of $q(u,v)=q_1(u)q_2(v)$ and of the signal design. We assume that each sender has both an average energy constraint and a bandwidth constraint on the signals his modulator can produce. Both a "second moment" and a "fractional out of band energy" constraint on bandwidth are considered. The optimizing modulator designs consist of a simplex signal set for each sender, such that each sender's set is orthogonal to that of the other's. For the second moment constraint, simplex sets derived from sinusoidal functions yield the optimal design, and for the alternative constraint, simplex sets derived from prolate spheroidal wave functions are optimal. We show for each bandwidth constraint that the cutoff rate region is not reduced compared to an unconstrained channel, but the code alphabet dimensions that can be accommodated are restricted. Achievable alphabet dimensions are identified as a function of the tightness of the bandwidth constraint.

---

*This work was supported by the National Science Foundation under Grant No. ENG-76-11565 and by the National Institutes of Health under Grant No. RR00396 from the Division of Research Resources.

For a Poisson process, additive-signal model, the received signal is a point process with count rate

$$\lambda(t) = s_1(t;i) + s_2(t;j) + \lambda_0,$$

where $s_1(t;i)$ is the signal count rate of sender-1 for symbol i, $s_2(t;j)$ is the signal count rate of sender-2 for symbol j, and λ_0 is the noise count rate, we also find that the cutoff rate region is maximized by independently maximizing each term in (1) as a function of $q(u,v) = q_1(u)q_2(v)$ and of the signal count rate design. We identify the optimizing design when each sender has both an average energy constraint and a peak amplitude constraint on the signals his modulator can produce.

1. J. L. Massey, "Coding and Modulation in Digital Communications," Proc. International Zurich Seminar on Digital Communications, Switzerland, March 1974.

2. D. L. Snyder and I. B. Rhodes, "Some Implications of the Cutoff Rate Criterion for Coded Direct Detection Optical Communication Systems," IEEE Trans. on Inform. Th., Vol. IT-26, pp. 327-338, May 1980.

PENETRATING TRANSMISSION ZEROS IN THE DESIGN OF ROBUST SERVOMECHANISM SYSTEMS

S.H. WANG
Department of Electrical Engineering
University of Maryland
College Park, Maryland 20742

E.J. DAVISON
Department of Electrical Engineering
University of Toronto
Toronto, Ontario
CANADA

ABSTRACT -- In the design of robust servomechanism system, it is well-known that a system can not track a reference signal whose frequency coincides with the transmission zeros of the system. In this paper, we propose a new design method to overcome this difficulty. The controller to be used employs sampler and holding device.

I. INTRODUCTION

The linear time-invariant servomechanism problem has been an important subject in control theory [1] - [6]. It is well-known that a system can not track a reference signal whose frequency coincides with the transmission zeros [7]-[11] of the system. For an excellent self-contained treatment on robust servomechanism problem, refer to [1]. In this paper, we propose a new design method to overcome this difficulty. The controller to be used employs sampler and holding device with exponential decay. It will be shown that the transmission zeros of the discretized system can be shifted by changing the rate of the exponential decay of the holding device. Therefore, one can design a robust controller for the discretized system to track any reference signal of given frequency, even if the given frequency coincides with the transmission zeros of the original continuous-time system. In this paper, we only consider single-input and single-output case. The multivariable case will be reported in a future paper. In Section II, our main theorem on shifting transmission zeros is given. In section III, we show by example how one can apply this result to robust servomechanism design. In the sequel, we use $\mathbb{R}$ to denote the set of real numbers and $\mathbb{C}$ to denote the set of complex numbers.

II. MAIN THEOREM

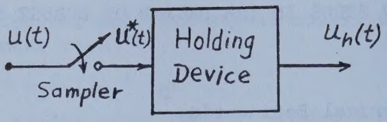
Consider a linear time-invariant single-input and single-output system specified by

$$\begin{aligned} \dot{x} &= Ax+bu \\ y &= cx \end{aligned} \tag{1}$$

Where $A = \text{diag}\,[\lambda_1,\ldots,\lambda_n]$, $b = (b_1,\ldots,b_n)'$, $c = (c_1,\ldots,c_n)$, and all λ's are real and distinct. We also assume that (A,b,c) is minimal, (i.e., completely controllable and completely observable). It should be noted that any minimal triple (A,b,c) can be transformed to the above form by state-feedback and state coordinate transformation. Consider the following sampler and holding device,

---

This research has been supported by National Aeronautics and Space Administration under Grant NAG1-27 and by the National Research Council of Canada under Grant A4396.

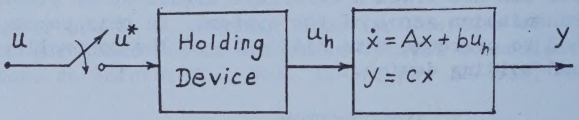


When the input signal u(t) is sampled at discrete instants 0,T,2T,..., where T is the sampling period, the holding device produces the signal $u_h(t)$, which is related to u(t) by the following equation:

$$u_h(t)=e^{-\alpha(t-kT)}u(kT),\ \text{for } kT\le t<(k+1)T. \tag{2}$$

Notice that this holding device reduces to the usual zero-order holding device when $\alpha=0$.

Consider the series connection of system (1) and the sampler and holding device in (2),



The following equations can be easily derived,

$$x((k+1)T) = e^{AT}x(kT) + e^{-\alpha T}\int_0^T e^{(A+\alpha I)t}dt\ b\ u(kT) \tag{3}$$

$$= \tilde{A}\ x(kT) + \tilde{b}\ u(kT).$$

$$y(kT) = \tilde{c}\ x(kT)$$

where $\tilde{A} = \text{diag}\ [e^{\lambda_1 T},\dots,e^{\lambda_n T}]$,

$$\tilde{b} = \left(\frac{e^{\lambda_1 T}-e^{-\alpha T}}{\lambda_1+\alpha}b_1,\dots,\frac{e^{\lambda_n T}-e^{-\alpha T}}{\lambda_n+\alpha}b_n\right)', \tag{4}$$

$\tilde{c} = c = (c_1,\dots,c_n)$, and $T > 0$.

The pulse transfer function for system (3) is

$$\tilde{h}(z)=\tilde{c}(zI-\tilde{A})^{-1}\tilde{b} = \frac{\tilde{n}(z)}{\det(zI-\tilde{A})}, \tag{5}$$

The following theorem states that by changing α, all transmission zeros of $(\tilde{A},\tilde{b},\tilde{c})$ can be shifted.

Theorem 1: Consider the triple $(\tilde{A},\tilde{b},\tilde{c})$ in (4) and its pulse transfer function $\tilde{h}(z)$ in (5). Then for all $z\in\mathbb{C}$, there exists $\alpha\in\mathbb{R}$, such that $\tilde{n}(z)\neq 0$. For a proof of Theorem 1, please refer to the Appendix.

III. EXAMPLE

Consider the following single-input and single-output system

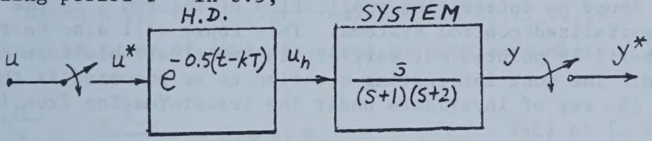
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 2 \end{bmatrix} u$$

$$y = \begin{pmatrix} 1 & 1 \end{pmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

The transfer function of the above system is

$$h(s) = c(sI-A)^{-1}b = \frac{s}{(s+1)(s+2)}.$$

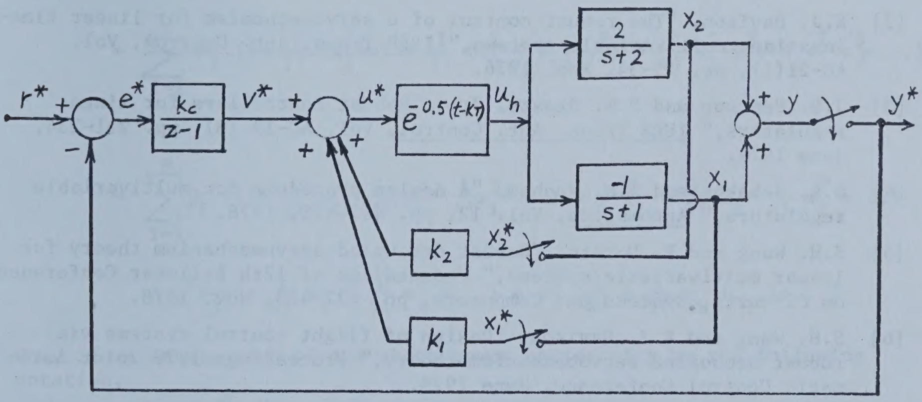
According to the robust servomechanism theory, this system can not track a step reference input, (see [1], pp. 8). Now, let us consider the series connection of a sampler and holding device with decay rate $\alpha = 0.5$, and sampling period $T = -\ln 0.5$,



Then the pulse transfer function from u^* to y^* is

$$\tilde{h}(z) = \frac{(0.19526)z - 0.2011825}{z^2 - 0.75z + 0.125} .$$

Notice that by choosing $\alpha = 0.5$, the transmission zero of the above discrete-time system is different from 1. Hence one can design a robust controller for the discrete-time system to track any step reference input at sampling instants. The complete configuration is shown below.



By choosing appropriate values for k_c, k_1 and k_2, one can assign arbitrary poles for the closed-loop system. For example, if we choose $k_c = -56.73$, $k_1 = 113.33$ and $k_2 = 75.63$, then the closed-loop poles are 0.2, 0.3 and 0.4.

Remark 1: It should be pointed out that our method per se may not be suitable for certain practical designs, because we have avoided pole-zero cancellation between controller and system by perturbing the system zeros, which may cause large gains for k_c, k_1 and k_2.

Remark 2: From another point of view, we have perturbed the system zeros in order to have the reference signal to penetrate the system, see [10], [11].

IV. CONCLUSIONS

In this paper, we have proposed a new method of designing robust servomechanism controllers. However, our method only guarantees that the system output will track reference input at sampling instants. The behavior of the system output y between sampling instants depends on the eigen-values λ_i's of the A matrix, which can be assigned by initial state-feedback, see remark following (1). This topic will be treated in another paper. Moreover, we have found an interesting application of Theorem 1 to the stabilization of decentralized control systems. This topic will also be reported in another paper. As pointed out earlier, the multivariable case has yet to be resolved. The most interesting question to be answered is the following: What is the set of invariants under the transformation from (A,b,c) in (1) to $(\tilde{A},\tilde{b},\tilde{c})$ in (3)?

ACKNOWLEDGEMENT

The authors are grateful to Professor R. Brockett for his helpful suggestions.

REFERENCES

[1] C.A. Desoer and Y.T. Wang, "Linear time-invariant robust servomechanism problem: A self-contained exposition", Memo. No. UCB/ERL M77/50, Electronics Research Laboratory, University of California, Berkeley, August 1977.

[2] E.J. Davison, "The robust control of a servomechanism for linear time-invariant multivariable systems," IEEE Trans. Aut. Control, Vol. AC-21(1), pp. 25-34, Feb. 1976.

[3] J.B. Pearson and P.W. Staats, Jr., "Robust controllers for linear regulators," IEEE Trans. Aut. Control, Vol. AC-19 (3), pp. 231-234, June 1974.

[4] O.A. Sebakhy and W.M. Wonham, "A design procedure for multivariable regulators," Automatica, Vol. 12, pp. 467-478, 1976.

[5] S.H. Wang and P. Dorato, "Robust decoupled servomechanism theory for linear multivariable systems," Proceedings of 12th Asilomar Conference on Circuits, Systems and Computers, pp. 427-430, Nov. 1978.

[6] S.H. Wang and E.J. Davison, "Design of flight control systems via robust decoupled servomechanism theory," Proceedings 1979 Joint Automatic Control Conference, June 1979.

[7] E.J. Davison and S.H. Wang, " Properties and calculation of transmission zeros of linear multivariable systems," Automatica, Vol. 10, pp. 643-658, 1974.

[8] S.H. Wang and E.J. Davison, "Canonical forms of linear multivariable systems," SIAM J. Contr. Optimiz., Vol. 14, pp. 236-250, Feb. 1976.

[9] E.J. Davison and S.H. Wang, "An algorithm for the calculation of transmission zeros of the system (A,B,C,D) using high gain output feedback," IEEE Trans. Aut. Control, Vol. AC-23 (4), pp. 738-741, August 1978.

[10] P.G. Ferreira and S.P. Bhattacharyya, "On blocking zeros," IEEE Trans. Aut. Control, Vol. AC-22 (2), pp. 258-259, April 1977.

[11] C.A. Desoer and J.D. Schulman, "Zeros and poles of matrix transfer functions and their dynamic interpretation," IEEE Trans. Circ. Sys., CAS-21, pp. 3-8, 1974.

APPENDIX

In order to prove Theorem 1, we need the following Lemma.

Lemma 1: Given a set of distinct real numbers $x_1, x_2, \ldots, x_n$. Let β_i, $(i=1,\ldots,n)$, be a set of arbitrary real or complex numbers. If

$$\sum_{i=1}^{n} \beta_i \left(\frac{e^{x_i}-e^{y}}{x_i-y}\right) = 0, \forall\, y \in \mathbb{R},$$

then $\beta_i=0$ for all $i \in \{1,2,\ldots,n\}$.

Proof: Replacing e^{x_i} and e^{y} by its Taylor's series about 0, we have

$$g(y) \triangleq \sum_{i=1}^{n} \beta_i \left(\frac{e^{x_i}-e^{y}}{x_i-y}\right)$$

$$= \sum_{i=1}^{n} \beta_i \left[1+\frac{1}{2!}(x_i+y)+\frac{1}{3!}(x_i^2+x_i y+y^2)+\frac{1}{4!}(x_i^3+x_i^2 y+x_i y^2+y^3)+\ldots\right]$$

$$= \sum_{i=1}^{n} \beta_i \left[\left(1+\frac{1}{2!}x_i+\frac{1}{3!}x_i^2+\frac{1}{4!}x_i^3+\ldots\right)+y\left(\frac{1}{2!}+\frac{1}{3!}x_i+\frac{1}{4!}x_i^2+\ldots\right)\right.$$

$$\left.+\,y^2\left(\frac{1}{3!}+\frac{1}{4!}x_i+\frac{1}{5!}x_i^2+\ldots\right)+y^3\left(\frac{1}{4!}+\frac{1}{5!}x_i+\ldots\right)+\ldots\right]$$

Now, if we denote each of the above power series of x by the following notation,

$$f_j(x) \triangleq \sum_{k=0}^{\infty} \frac{x^k}{(k+j+1)!}, \quad j=0,1,2,\ldots,$$

then

$$g(y)=\sum_{i=1}^{n}\beta_i\Big(\sum_{j=0}^{\infty} y^j f_j(x_i)\Big)$$

$$=\sum_{j=0}^{\infty} y^j\Big(\sum_{i=1}^{n}\beta_i f_j(x_i)\Big)$$

Since g(·) has derivative of every order at each point in $\mathbb{R}$, and $g(y)=0$ for all $y\in\mathbb{R}$, we have

$$\frac{d^m}{dy^m}g(y)\Big|_{y=0}=m!\sum_{i=1}^{n}\beta_i f_m(x_i)$$

$$=m!\sum_{i=1}^{n}\beta_i\Big(\sum_{k=0}^{\infty}\frac{x_i^k}{(k+m+1)!}\Big)$$

$$=\frac{1}{(m+1)}\Big[\sum_{i=1}^{n}\beta_i+\frac{1}{(m+2)}\sum_{i=1}^{n}\beta_i x_i+\frac{1}{(m+2)(m+3)}\sum_{i=1}^{n}\beta_i x_i^2+\ldots\Big]$$

$$=\frac{1}{(m+1)}\Big[\sum_{i=1}^{n}\beta_i+\sum_{k=1}^{\infty}\frac{1}{\prod_{j=1}^{k}(j+m+1)}\Big(\sum_{i=1}^{n}\beta_i x_i^k\Big)\Big] \qquad \text{(A1)}$$

$= 0$, for all $m\geq 0$.

Consider the second term in the above parenthesis,

$$\left|\sum_{k=1}^{\infty}\frac{1}{\prod_{j=1}^{k}(j+m+1)}\Big(\sum_{i=1}^{n}\beta_i x_i^k\Big)\right|\leq\frac{n\beta x}{m+2}\sum_{k=0}^{\infty}\frac{x^k}{k!}=\frac{n\beta e^x x}{m+2} \qquad \text{(A2)}$$

where $\beta=\max_i|\beta_i|$ and $x=\max_i|x_i|$.

From (A2), when $m \to \infty$, the second term in (A1) goes to zero. Since the sum of the two terms in (A1) is zero for all $m > 0$, the first term of (A1) must be zero, i.e.,

$$\sum_{i=1}^{n} \beta_i = 0.$$

By the same procedure, we can show that

$$\sum_{i=1}^{n} \beta_i x_i^k = 0, \quad \text{for all } k \in \{1,2,\ldots,n-1\}.$$

From the above two equations,

$$[\beta_1, \ldots, \beta_n] \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^{n-1} \\ 1 & x_2 & x_2^2 & \cdots & x_2^{n-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & x_n & x_n^2 & & x_n^{n-1} \end{bmatrix} = [0, \ldots, 0]$$

Since all x_i's are distinct, the above square matrix is nonsingular. Hence, $\beta_i = 0$ for all $i \in \{1,2,\ldots,n\}$. Q.E.D.

Proof of Theorem 1: We shall prove by contradiction. Assume there exists $z \in \mathbb{C}$, such that for all $\alpha \in \mathbb{R}$,

$$\tilde{n}(z) \equiv \sum_{i=1}^{n} \left[\prod_{\substack{j=1 \\ j \neq i}}^{n} (z - e^{\lambda_j T}) \right] c_i b_i T \left(\frac{e^{\lambda_i T} - e^{-\alpha T}}{\lambda_i T + \alpha T} \right) = 0.$$

From Lemma 1, we have

$$\left[\prod_{\substack{j=1 \\ j \neq i}}^{n} (z - e^{\lambda_j T}) \right] c_i b_i T = 0, \text{ for all } i \in \{1,\ldots,n\} .$$

Since it is assumed that (A,b,c,) is minimal and $T > 0$, we have $c_i b_i T \neq 0$.

Hence

$$\prod_{\substack{j=1 \\ j \neq i}}^{n} (z - e^{\lambda_j T}) = 0 \text{ for all } i \in \{1,\ldots,n\} .$$

This is impossible because we have assumed that all λ_i's are real and distinct. Q.E.D.

ON A DISTURBANCE COMPENSATION METHOD*

J. KIM and D. H. CHYUNG
Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242

ABSTRACT

A disturbance compensation method for reducing the effects of external disturbance in a third order position control system is investigated.

One of the problems encountered by a position control system on a moving platform is the error due to the disturbance caused by the movement of the platform. In this paper, a compensation method for reducing the effects of the disturbance in a d.c. motor driven third order position control system is investigated. Figure 1 below shows the system under study. The

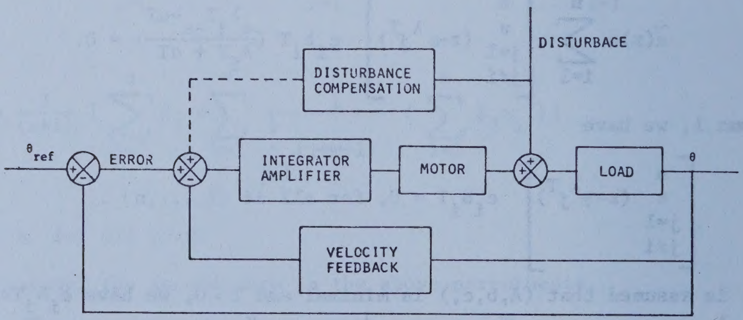

DISTURBACE
DISTURBANCE COMPENSATION
θref
ERROR
INTEGRATOR AMPLIFIER
MOTOR
LOAD
θ
VELOCITY FEEDBACK
+
-

Fig.1 Position Control System

---

*This research was supported in part by NSF Grant No. ENG78-00814 and U.S. Army Grant No. DAAK10-78-C-0325.

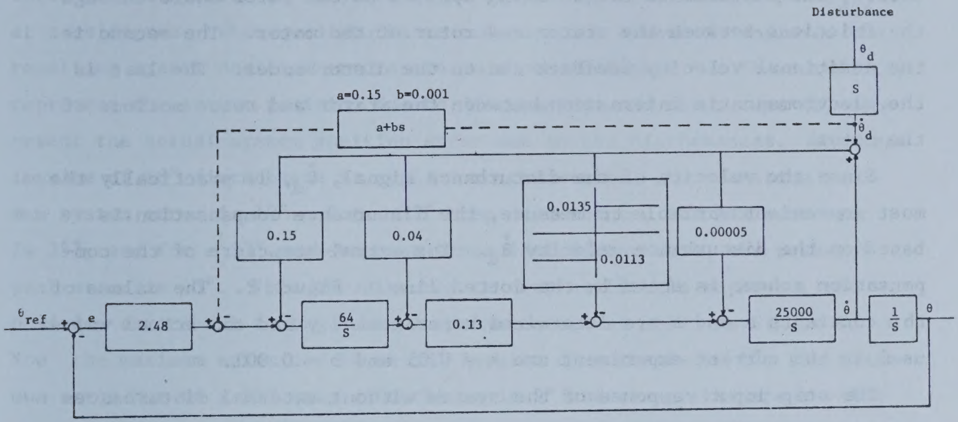


Fig.2 Block Diagram

block diagram model of the system is as shown in Figure 2. As can be seen from the block diagram, there is a substantial nonlinear Coulomb friction between the stator and rotor of the d.c. motor. When a typical second order position control, that is, without the additional integrator in Figure 1, is employed, there is a large steady state error due to the nonlinear friction. The additional integrator is therefore inserted in the system to force the steady state error to be zero. Another reason for the inclusion of the integrator is to investigate the effectiveness of the proposed disturbance compensation scheme in a third order system.

To simulate the disturbance due to the motion of a platorm, the stator, that is, the outer shell, of the motor is rotated, and the effects of the stator motion on the position error of the system is studied. The disturbance motion of the platform, and therefore, the motion of the motor stator, is denoted by θ_d, and the system position, that is, the position of the motor rotor is denoted by θ. The system error is defined by

$$e = \theta_{ref} - \theta$$

where θ_{ref} is the desired position angle. The disturbance stator motion θ_d affects the rotor position θ in three major ways. The first is, of course, the disturbance torque being applied to the rotor shaft through the frictions between the stator and rotor of the motor. The second is the additional velocity feedback due to the disturbances. The last is the electromagnetic interaction between the stator and rotor motions of the motor.

Since the velocity of the disturbance signal, $\dot{\theta}_d$, is practically the most convenient variable to measure, the disturbance compensation is based on the disturbance velocity $\dot{\theta}_d$. The actual structure of the compensation scheme is shown by the dotted line in Figure 2. The values of the constants a and b are determined experimentally and the actual values used in the current experiment are a = 0.15 and b = 0.001.

The step input response of the system without external disturbances is as shown in Figure 3. The settling time is about 0.8 sec. in one direction and 2.6 sec. in the other direction. The difference in settling time in the positive and negative directions is due to unsymmetrical nonlinear frictions.

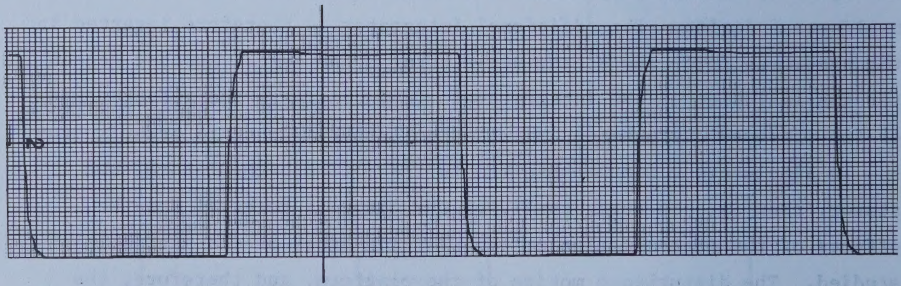
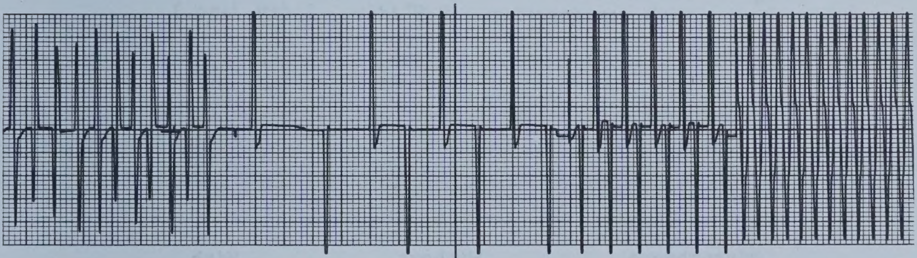
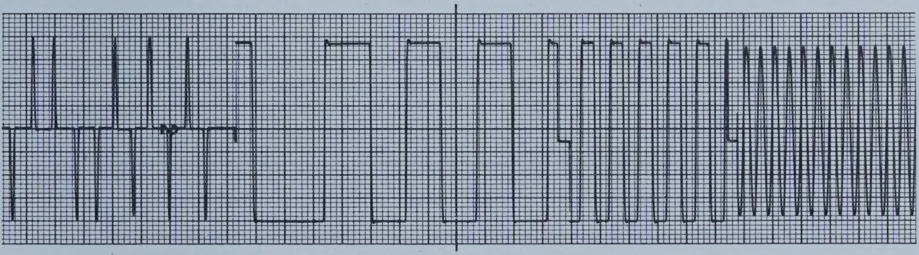


Fig. 3 Step Input Response (0.9°/div., 5 div./sec.)

To study the effects of the disturbances, three different types of disturbance motions are applied to the stator of the motor, and the rotor position error is observed while keeping the reference input θ_{ref} at zero degree. When no disturbance compensation scheme is employed the resulting system position errors are as shown in Figure 4. The curves (a) represent the stator position due to disturbances and the curves (b) represent the actual system position error due to the disturbances. When an impulse of 18° disturbance is applied to the stator, the resulting maximum system error is 19°. For a 35° step disturbance, the maximum error is 25°, and for a 33° peak-to-peak sinusoidal disturbance, the peak-to-peak error is 45°. When the disturbance compensation is employed, the position errors due to the same disturbances are as shown in Figure 5. Now the maximum error due to an 18° impulse disturbance is 6°, the maximum error due to a 35° step disturbance is 9° and the peak-to-peak value of the error due to 35° p-p sinusoidal disturbance is 14°. The results are summerized in Table 1. In general, the disturbance compensation scheme reduces the effects of external disturbances significantly, in the present case, by a factor of 3.

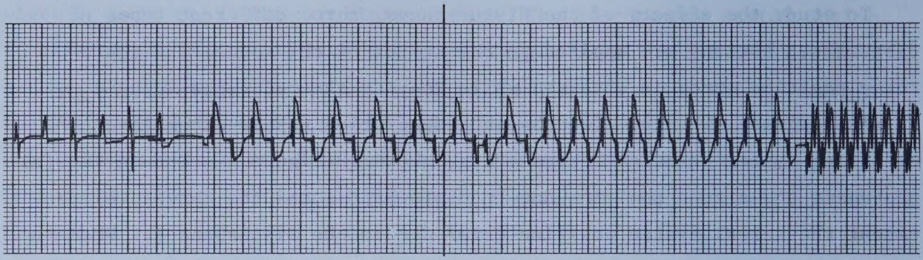


(a) Position error ($\theta-\theta_{ref}$)

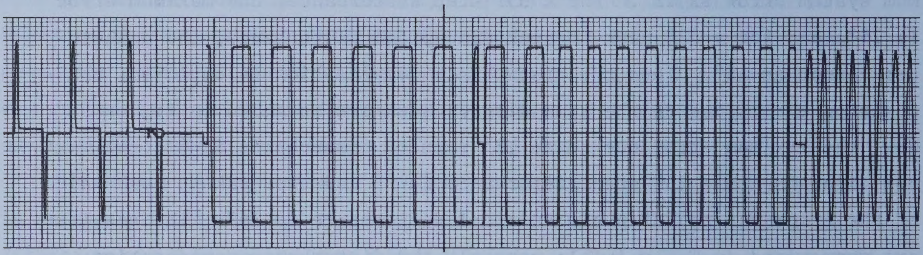


(b) Disturbance input (θ_d)

Fig. 4 Position error due to disturbance in the system without compensation (0.9°/div., 5 div./sec.)



(a) Position error $(\theta-\theta_{ref})$



(b) Disturbance input (θ_d)

Fig. 5 Position error due to disturbance in the system with compensation (0.9°/div., 5 div./sec.)

Disturbance input	Without compensation	With compensation
18° Impulse	19°	6°
35° Step	25°	9°
33° p-p 2Hz.	45° p-p	14° p-p

Table 1. Position error due to external disturbances

In conclusion, it is shown that a disturbance compensation scheme may be employed to reduce the effects of external disturbances in a position control system. Since the major portion of the disturbance is transmitted to the control system through the nonlinear friction, a nonlinear disturbance compensation scheme would be more effective. Further research is currently being carried out to more fully develop the disturbance compensation methods.

DYNAMIC INVERSE OF GENERALIZED STATE SPACE SYSTEMS

V. A. OLIVEIRA
Control Systems Group
Dept. of Electrical Engineering
COPPE, Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil

S. P. BHATTACHARYYA
Dept. of Electrical Engineering
Texas A & M University
College Station, Texas 77843, USA

ABSTRACT

This paper deals with the linear discrete time dynamic system

$$Tx_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, 2, \ldots \tag{1}$$

with T being singular. Such systems are known as generalized state space systems [1] and are of current interest [2], [3]. An algorithm is described in this paper to place the equations of the above system in recursive form. This is an important step in the analysis and synthesis of such systems. The algorithm given here is a straightforward adaptation of Silverman's system inversion algorithm [4] and differs from the solution given in [2]. The algorithm also provides structural information on (1) and should prove useful in synthesis.

REFERENCES

[1] P. Van Dooren, G. Verghese and T. Kailath, "Properties of the system matrix of generalized state space systems," Proc. Conf. on Decision and Control, San Diego, CA, 1978.

[2] D. G. Luenberger, "Time invariant description systems," Automatica, Vol. 14, pp. 473-480, 1978.

[3] J. D. Aplevich, "A system representation for control digital signal processing and estimation," Proc. IEEE, Vol. 67, No. 11, pp. 1557-1559, Nov. 1979.

[4] L. M. Silverman, "Inversion of multivariable linear systems," IEEE T-AC, Vol. AC-14, No. 3, pp. 270-276, June 1969.

POLE ASSIGNMENT COMPENSATORS WITH ARBITRARY SPECTRUM

S. P. BHATTACHARYYA
Dept. of Electrical Engineering
Texas A & M University
College Station, Texas 77843

ABSTRACT

It is shown that for the linear time invariant controllable, observable system

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

the dynamic compensator

$$\dot{x}_c = A_c x_c + B_c y + G_c u$$

$$u = C_c x_c + D_c y$$

can always be designed to assign a prescribed closed loop spectrum with A_c having a prescribed spectrum. The implications of this fact on the stability and construction of stabilizing compensators and observers are discussed.

A NEW RESULT FOR OUTPUT FEEDBACK DECOUPLING

H. Y. KIM AND E. Y. SHAPIRO

Lockheed-California Company
Burbank, California

ABSTRACT

A new necessary and sufficient condition is derived to decouple linear time-independent multivariable systems through output feedback. As a consequence of this result an easily computable, sufficient condition for output feedback decoupling is obtained, which is expressed in terms of the open loop parameters of the system. This condition is a considerable improvement over the one recently derived by Sinha in [7].

I. INTRODUCTION

Since the fundamental work of Falb and Wolovich [1], Gilbert [2], and Wonham and Morse [3] on decoupling of multivariable linear systems using state feedback, considerable attention has been paid by many authors to the decoupling problem. As the full state is, in many practical cases, not always available, it is natural to turn their attention to the problem of output feedback decoupling [4] - [6].

In [5] Wolovich developed an elegant condition for output feedback decoupling in the frequency domain approach. However, the inversion of the transfer matrix of the open-loop system, which is required in his approach, is not so simple in practice. The main objective of this paper is to derive a new characterization of output feedback decoupling which can easily be implemented as a software package.

II. DECOUPLING CONTROL LAWS

Consider a linear time-invariant multivariable system $S = \{A, B, C\}$:

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx \tag{2}$$

where $x \epsilon R^n$, $y \epsilon R^m$, $u \epsilon R^m$, and A, B, C are matrices of appropriate sizes. The system is said to be decouplable by the state feedback control law

$$u = Fx + Gv \tag{3}$$

where $v \epsilon R^m$ and F and G are matrices of sizes mxn and mxm respectively, if the resulting system $S(F,G) = \{A + BF, BG, C\}$ has a nonsingular diagonal transfer matrix.

Define, for each i, a non-negative integer d_i, by

$$d_i = \min\left\{n-1, \{j \mid C_i A^j B \neq 0\}\right\} \tag{4}$$

where C_i is the ith row of C, and let mxm matrix

$$D = \begin{pmatrix} C_1 A^{d_1} B \\ \cdot \\ \cdot \\ \cdot \\ C_m A^{d_m} B \end{pmatrix} \tag{5}$$

Then the system S is decouplable by a matrix pair $\{F, G\}$ if and only if D is nonsingular [1].

It will be assumed throughout this correspondence that the matrix D is nonsingular. Also assumed is the controllability of the system S except in IV. Define a new system $\bar{S} = \{\bar{A}, \bar{B}, \bar{C}\}$ by

$$\bar{A} = A - B D^{-1} A^{*} \tag{6}$$

$$\bar{B} = B D^{-1} \tag{7}$$

$$\bar{C} = C \tag{8}$$

where

$$A^{*} = \begin{pmatrix} C_1 A^{d_1+1} \\ \cdot \\ \cdot \\ \cdot \\ C_m A^{d_m+1} \end{pmatrix} \tag{9}$$

It is easy to show $d_i = \bar{d}_i$ where $\bar{d}_i$ is the integer defined by (4) for the system $\bar{S}$. Let

$$W_i = \left\{ \eta \in R^n \;\middle|\; \eta \bar{A}^j \bar{B}_k = 0,\ j = 0, 1, \ldots, n-1;\ k \neq i \right\} \tag{10}$$

where $\bar{B}_k$ is the kth column of $\bar{B}$. (The dual of R^n is identified with itself in (10).) Then the following lemma is known [2].

Lemma 1: The subspace W_i has the properties:

i) W_i is $\bar{A}$-invariant,

ii) $W_i \cap W_j = 0$, $i \neq j$,

iii) $C_i, C_i \bar{A}, \ldots, C_i \bar{A}^{d_i}$ are linearly independent vectors belonging to W_i.

iv) $W_i \cap \left\{ W_1 + \ldots + W_{i-1} + W_{i+1} + \ldots + W_m \right\} = \left\{ 0 \right\}$

Let p_i, i=1, 2, ..., m, be the dimension of W_i and $\left\{ C_i, C_i \bar{A}, \ldots, C_i \bar{A}^{d_i}, q_1^i, \ldots, q_{r_i}^i \right\}$ a basis of W_i where $r_i = p_i - d_i - 1 \geq 0$, and define $p_i \times n$ matrix Q_i by

$$Q_i = \begin{pmatrix} C_i \\ C_i \bar{A} \\ \cdot \\ \cdot \\ \cdot \\ C_i \bar{A}^{d_i} \\ q_1^i \\ \cdot \\ \cdot \\ \cdot \\ q_{r_i}^i \end{pmatrix} \tag{11}$$

Because of (iv) of Lemma 1 there exists a $(p_{m+1} \times n)$ matrix Q_{m+1}, where $p_{m+1} = n - (p_1 + \ldots + p_m)$, such that

$$Q = \begin{pmatrix} Q_1 \\ \cdot \\ \cdot \\ \cdot \\ Q_m \\ Q_{m+1} \end{pmatrix} \tag{12}$$

is a nonsingular matrix. With this matrix the set of decoupling pairs $\{F, G\}$ is characterized by [2]

Theorem 1: A matrix pair $\{F, G\}$ decouples the system S if and only if

$$F = -D^{-1} A^{*} + D^{-1} \begin{pmatrix} \theta_1 & 0 & 0 \dots 0 & 0 \\ 0 & \theta_2 & 0 \dots 0 & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 \dots \theta_m & 0 \end{pmatrix} Q \tag{13}$$

and

$$G = D^{-1} \operatorname{diag} (\lambda_1, \dots, \lambda_m), \lambda_i \neq 0, i = 1, 2, \dots, m, \tag{14}$$

where θ_i is 1 x p_i.

III. MAIN RESULTS

In view of (1), (2), (3), (13), (14) output feedback

$$u = Hy + Gv \tag{15}$$

decouples the system if and only if mxn matrix HC is in the form of (13) and G is given by (14).

Theorem 2: Let C be the subspace of R^n spanned by vectors $C_1, C_2, \dots, C_m$. Then S is decoupled through output feedback if and only if

$$A_i^{*} \in C + W_i , i = 1, 2, \dots, m, \tag{16}$$

where A_i^{*} is the ith row of A^{*}.

Proof: If $A_i^{*} \in C + W_i$, then represent A_i^{*} in the form of

$$-A_i^{*} = \sum_{j=1}^{m} h_{ij} C_i + \sum_{j=0}^{d_i} a_{ij} C_i \bar{A}^j + \sum_{j=1}^{r_i} \beta_{ij} q_j^i \tag{17}$$

Let $\mathcal{F}$ be the set of all mxn matrices of the form (13). With $H = D^{-1}(h_{ij})$ it is easy to see $HC \in \mathcal{F}$. Suppose $HC \in \mathcal{F}$. Then for some θ_{ij}, $j = 1, 2, \dots, p_i$, $i = 1, 2, \dots, m$,

$$-A_i^{*} + \sum_{j=0}^{d_i} \theta_{ij} C_i \bar{A}^j + \sum_{j=1}^{r_i} \theta_{i,j+d_i+1} q_j^i = \sum_{j=1}^{m} h_{ij} C_j \tag{18}$$

where $\{h_{ij}\}_{j=1}^{m}$ is the ith row of the matrix DH. This implies $A_i^{*} \in C + W_i$.

It should be observed that, as $C \cap W_i = sp\{C_i\}$ where $sp\{C_i\}$ denotes the subspace spanned by C_i (see Lemma 1), the representation of A_i^{*} in the form of (17) is not unique. It is easy to see that, if the system is decouplable through output feedback, there are exactly m free parameters. This assertion is valid even for systems which are not controllable. (See Theorem 3 in IV.)

Corollary 1: Let V_i, i = 1, 2, ..., m, be any subspace of W_i. Then

$$-A_i^{*} \in C + V_i , i = 1, 2 \dots, m, \tag{19}$$

is a sufficient condition for output feedback decoupling.

One apparent drawback of the theorem is that W_i is not directly related to the original parameters as it is defined through the matrices $\bar{A}$, $\bar{B}$. To find a direct link take

$$V_i = sp \left\{ C_i, C_i \bar{A}, \dots, C_i \bar{A}^{d_i} \right\} \tag{20}$$

in Corollary 1. In view of the obvious relation

$$C_i A^j = C_i \bar{A}^j, \; j = 0, 1, \dots, d_i, \; i = 1, 2, \dots, m, \tag{21}$$

the following sufficient condition is obtained in which only open loop parameters are used.

Corollary 2: If, for each i = 1, 2, ..., m, the ranks of the matrices

$$\begin{pmatrix} C_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ C_m \\ C_i A \\ \cdot \\ \cdot \\ \cdot \\ C_i A^{d_i} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} C_1 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ C_m \\ C_i A \\ \cdot \\ \cdot \\ \cdot \\ C_i A^{d_i+1} \end{pmatrix} \tag{22}$$

are same, the system can be decoupled through output feedback.

Let, for i = 1, 2, ..., m,

$$T_i = \begin{pmatrix} C_i \\ C_i A \\ \cdot \\ \cdot \\ \cdot \\ C_i A^{d_i+1} \end{pmatrix} \tag{23}$$

and define the matrix T by

$$T = \begin{pmatrix} T_1 \\ T_2 \\ \cdot \\ \cdot \\ \cdot \\ T_m \end{pmatrix} \tag{24}$$

It is shown in [7] that if the rank of the matrix T is same as that of the matrix C then the system is decouplable using output feedback. To study this problem a little more carefully the following lemma is useful.

Lemma 2: Suppose $C_i A^{d_i} \in C$ for some i with $d_i > 0$. Then the system is not decouplable through full state feedback. (Note that this statement is true even if the system is not controllable.)

Proof: Let J_o be the subset of $\{1, 2, \dots, m\}$ defined by

$$J_o = \{ j \mid d_j = 0 \} . \tag{25}$$

If $C_i A^{d_i} \in C$

$$C_i A^{d_i} = \sum_{j=1}^{m} \alpha_{ij} C_j \tag{26}$$

and hence

$$C_i A^{d_i} B = \sum_{j=1}^{m} \alpha_{ij} C_j B = \sum_{j \in J_o} \alpha_{ij} C_i B \tag{27}$$

As $i \neq J_o$ by assumption this shows that ith row of D consists of a linear combination of other rows of D which tells that the matrix D is singular.

By this lemma, if rank (T) = m, and the system S is decouplable (i.e., D is nonsingular), $d_i = 0$ for all i = 1, 2, ..., m. Then the rank condition of [7] is equivalent to

$$C_i A^{d_i+1} = C_i A = A_i^* \in C, \ i = 1, 2, \ldots, m. \tag{28}$$

Clearly this is a special case of Corollary 1 with $V_i = \{o\}$. It is not surprising, in view of the above argument, that example 1 of [7] has $d_1 = d_2 = 0$ and rank condition is not satisfied in example 2 of [7] as $d_1 = 1 > 0$. This lemma is not entirely unexpected in view of the fact that $C_i \in W_i$, $W_i \cap \sum_{j\neq 1} W_j = \{0\}$, $C_i A^{d_i} \in W_i$, and $C_i A^{d_i}$ and C_i are linearly independent vectors if $d_i > 0$. (See Lemma 1.)

IV. NONCONTROLLABLE SYSTEMS

Suppose now the system S is not controllable. Thus the dimension n_c of the subspace spanned by $\{A^j B_k \mid j = 0, 1, \ldots, n-1, k = 1, 2, \ldots, m\}$ is strictly less than the order n of the system. It is well known in this case that the controllable part can be separated from the uncontrollable part as manifested by:

Lemma 3: Let $S = \{A, B, C\}$ be an nth order system which is not controllable. Let n_c be the dimension of the subspace spanned by $\{A^j B_k \mid j = 0, 1, \ldots, n-1, k = 1, 2, \ldots, m\}$. Then a suitable nonsingular linear transformation of the state

$$\tilde{x} = T_1 x \tag{29}$$

yields an equivalent system $\tilde{S} = \{\tilde{A}, \tilde{B}, \tilde{C}\}$ where

i)

$$\tilde{A} = \begin{pmatrix} A^c & A_1^u \\ 0 & A_2^u \end{pmatrix} \tag{30}$$

$$\tilde{B} = \begin{bmatrix} B^c \\ 0 \end{bmatrix} \tag{31}$$

$$\tilde{C} = \left[C^c , C^u \right] \tag{32}$$

where A^c, B^c, C^c are $n_c \times n_c$, $n_c \times m$, $m \times n_c$ matrices,

ii) $S^c = \{A^c, B^c, C^c\}$ is controllable.

iii) If S is integrator decoupled (ID), so is S^c (See [2] for the definition and properties of ID system.)

Transformation of the system $\bar{S}$ of (6) - (8) yields a new controllable system S^c. As $\bar{S}$ is ID, so is S^c, and the results obtained above for $\bar{S}$ are then applicable to S^c. The notations developed before will be used here with superscript c or tilde, e.g., C_i^c, W_i^c, d_i^c etc. Observe that for each i = 1, 2, ..., m,

$$d_i = \tilde{d}_i \tag{33}$$

as T_1 is nonsingular matrix. Furthermore in view of zero blocks in $\tilde{A}$ and $\tilde{B}$, it is easy to show that

$$d_i^c = \tilde{d}_i \tag{34}$$

It is proved in [2] that a pair $\{F, G\}$ decouples the system S if and only if

$$F = -D^{-1} A^{*} + D^{-1} \begin{bmatrix} \theta_1 & 0 & 0 & \dots & 0 & 0 & \theta_1^u \\ 0 & \theta_2 & 0 & \dots & 0 & 0 & \theta_2^u \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & & \theta_m & 0 & \theta_m^u \end{bmatrix} T_2 T_1 \tag{35}$$

and

$$G = D^{-1} \operatorname{diag} (\lambda_1, \dots, \lambda_m), \; \lambda_i \neq 0, \; i = 1, 2, \dots, m. \tag{36}$$

where

$$T_2 = \begin{bmatrix} Q^c & 0 \\ 0 & I_{n-n_c} \end{bmatrix} \tag{37}$$

and

$$\theta_i \text{ is } 1 \times p_i^c, \text{ and } \theta_i^u \text{ is } 1 \times (n-n_c), \; i = 1, 2, \dots, m. \tag{38}$$

The problem of output feedback decoupling is then reduced to:

Find an m x m matrix H such that

$$D^{-1} (-A^{*} + \Theta T_2 T_1) = HC \tag{39}$$

or

$$D^{-1} (-A^{*} T_1^{-1} + \Theta T_2) = HCT_1^{-1} \tag{40}$$

where Θ is the matrix with θ_i's and θ_i^u's in (35).

Theorem 3: Let $\hat{A}$ be the first n_c columns of m x n matrix $A^{*} T_1^{-1}$. Then

$$\hat{A}_i \in C^c + W_i^c , \; i = 1, 2, \dots, m \tag{41}$$

is a necessary and sufficient condition for output feedback decoupling, where

$$C^c = \operatorname{sp} \left\{ C_1^c, C_2^c, \dots, C_m^c \right\} \tag{42}$$

and

$$W_i^c = \left\{ \eta \in R^{n_c} \;\middle|\; \eta (A^c)^j B_k^c = 0, \; j = 0, 1, \dots, n_c, \; k \pm i \right\} \tag{43}$$

Proof. The necessity is obvious as

$$T_2 = \left(\begin{array}{c|c} Q_1^c & \\ \cdot & \\ \cdot & 0 \\ \cdot & \\ Q_{m+1}^c & \\ \hline 0 & I_{n-n_c} \end{array} \right) \tag{44}$$

and a special form Θ of (40) takes (Note that $CT_1^{-1} = \tilde{C} = (C^c, C^u)$).

Suppose now (41) is true. Then there exists an m x m matrix H which satisfies first n_c columns of (40). As the last $(n-n_c)$ columns of Θ are arbitrary, suitable choice of these $(n-n_c)$ columns of Θ can be made to satisfy (40).

Corollary 3: Let V_i be the subspace as given in (20), for each i = 1, 2, ..., m. Then if

$$A_i^* \in C + V_i \tag{45}$$

there exists an output feedback decoupling of the system S.

Proof: If

$$A_i^* \in C + V_i \tag{46}$$

then

$$A_i^* = \sum_{j=1}^{m} \alpha_{ij} C_j + \sum_{j=0}^{d_i} \beta_{ij} C_i A^j \tag{47}$$

and

$$A_i^* T_1^{-1} = \sum_{j=1}^{m} \alpha_{ij} \tilde{C}_j + \sum_{j=0}^{d_i} \beta_{ij} \tilde{C}_i \tilde{A}^j \tag{48}$$

as $C_i A^j = \bar{C}_i \bar{A}^j$ and so

$$C_i A^j T_1^{-1} = \tilde{C}_i \tilde{A}^j , \; j = 0, 1, \ldots, d_i. \tag{49}$$

Clearly (48) implies (41) since $\tilde{A}^j$ has the form of

$$\tilde{A}^j = \begin{pmatrix} (A^c)^j & X \\ 0 & (A_2^u)^j \end{pmatrix} \tag{50}$$

where X in (50) denotes an irrelevant submatrix of $\tilde{A}^j$ for the first n_c columns of $\tilde{C}_i \tilde{A}^j$.

Thus, even for noncontrollable systems Corollary 2 is applicable.

It appears that the application of Theorem 3 is rather involved as the transformation matrix T_1 of (29) as well as its inverse is apparently required in the process. However, this can be easily avoided as the following lemma manifests.

Lemma 4: Let L be an n x n_c matrix consisting of basis vectors of the subspace spanned by the mn columns $\bar{A}^j \bar{B}_k$, j = 0, 1, ..., n-1, k = 1, 2, ..., m. Then W_i^c is characterized by

$$W_i^c = \left\{ \eta^c \,\middle|\, \eta^c = \xi L , \; \xi \in V_i \right\} \tag{51}$$

where

$$V_i = \left\{ \xi \,\middle|\, \xi \bar{A}^j \bar{B}_k = 0, \; j = 0, 1, \ldots, n_c-1, \; k \neq i \right\} \tag{52}$$

Proof: Let $\xi \in V_i$. Then by definition $\xi \bar{A}^j \bar{B}_k = 0$, j = 0, 1, ..., n_c-1, k ≠ i. As

$$\xi \bar{A}^j \bar{B}_k = \xi T_1^{-1} \tilde{A}^j \tilde{B}_k = \xi T_1^{-1} \begin{pmatrix} (A^c)^j B_k^c \\ 0 \end{pmatrix} \tag{53}$$

$$\left(\xi T_1^{-1} \right)_{n_c} (A^c)^j B_k^c = 0. \tag{54}$$

It is well known that the first n_c columns of T_1^{-1} can be taken to be L.

Thus $\left(\xi T_1^{-1} \right)_{n_c} = \xi L$ and $\xi L \in W_i^c$. Suppose $\eta^c \in W_i^c$. Then for any 1 x (n - n_c) η^u

$$(\eta^c, \; \eta^u) \begin{pmatrix} (A^c)^j B_k^c \\ 0 \end{pmatrix} = 0 \tag{55}$$

i.e.

$$(\eta^c , \eta^u) \ \tilde{A}^j \ \tilde{B}_k = 0 \tag{56}$$

For $\xi = (\eta^c, \eta^u) T_1$

$$\xi \bar{A}^j \bar{B}_k = 0 \tag{57}$$

which shows that $\xi \epsilon V_i$ and $\eta^c = \xi L$.

Lemma 4 suggests a way to find W_i^c: Solve $Hx = 0$ where H is the matrix consisting of row vectors $\bar{B}_k^T (\bar{A}^T)^j$, $j = 0, 1, \ldots, n_c - 1$, $k \neq i$. Then $X^T L$ is a vector in W_i^c. It should be observed that $C_i^c (A^c)^j = \bar{C}_i (\bar{A})^j L$ $j = 0, 1, \ldots, d_i$, are in W_i^c.

V. EXAMPLES

Example 1: Let

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 1 \\ -1 & 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{58}$$

Then

$$D = \begin{pmatrix} -2 & 2 \\ 1 & 0 \end{pmatrix} \tag{59}$$

$$A^* = \begin{pmatrix} 3 & -3 & 0 \\ -1 & 1 & 0 \end{pmatrix} \tag{60}$$

$$Q_1 = \begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & -2 \end{pmatrix} , \quad Q_2 = (0 \quad 0 \quad 1) \tag{61}$$

As

$$A_1^* = 3C_1 \quad , \quad A_2^* = -C_1 \tag{62}$$

the system is decouplable through output feedback. From (17), (61) it is easy to see

$$\begin{array}{ll} h_{12} = \theta_{12} = 0 \quad , & h_{11} + \theta_{11} = -3 \\ h_{21} = 1 & h_{22} + \theta_{21} = 0 \end{array} \tag{63}$$

and so the matrix H is given by

$$H = D^{-1} (h_{ij}) = \begin{pmatrix} 1 & -\theta_{21} \\ \dfrac{-\theta_{11} - 1}{2} & -\theta_{21} \end{pmatrix} \tag{64}$$

where θ_{11} and θ_{21} are arbitrary numbers. The transfer matrix $T_d(s)$ of the closed-loop system is then

$$T_d(s) = C \ (sI - A - BHC)^{-1} BG = \begin{pmatrix} \dfrac{\lambda_1}{s^2 + \theta_{11}} & 0 \\ 0 & \dfrac{\lambda_2}{s + \theta_{21}} \end{pmatrix} \tag{65}$$

as

$$G = D^{-1} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{66}$$

Wolovich derived in [5] elegant results for output feedback decoupling in the frequency domain approach. His theorem applied to the present example yields the following:

As the transfer function T(s) of the open loop system is

$$T(s) = C \ (sI - A)^{-1} \ B = \frac{1}{d(s)} \begin{pmatrix} -2s & 2s \\ s^2 - 1 & -2 \end{pmatrix} \tag{67}$$

where $d(s) = s(s^2 - 3)$

$$D \ T(s)^{-1} = \begin{pmatrix} s^2 - 3 & 0 \\ 1 & s \end{pmatrix} \tag{68}$$

Thus the system is decouplable through output feedback, since the off-diagonal entries of $D\,T(s)^{-1}$ are all constants, and the closed-loop transfer matrix $T_d(s)$ is

$$T_d(s) = \begin{pmatrix} \dfrac{\lambda_1}{s^2 - \hat{h}_{11} - 3} & 0 \\ 0 & \dfrac{\lambda_2}{s - \hat{h}_{22}} \end{pmatrix} \tag{69}$$

where $\hat{h}_{11}$ and $\hat{h}_{22}$ are arbitrary numbers. (See [5] for detail.) This is in agreement with (65) since $\theta_{11} = -3 - \hat{h}_{11}$, $\theta_{21} = -\hat{h}_{22}$ (See (63).) Note that (h_{ij}) in (63) is denoted by $\hat{H}$ in [5].

Example 2: (Falb and Wolovich [1]. See also [7].)

Let

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 1 & 3 \end{pmatrix} \qquad B = \begin{pmatrix} 1 & 1 \\ -1 & 1 \\ 0 & 0 \end{pmatrix} \qquad C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{70}$$

Then

$$D = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \tag{71}$$

$$A^* = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 5 & 9 \end{pmatrix} \tag{72}$$

$$Q_1 = (1 \quad 0 \quad 0), \; Q_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 3 \end{pmatrix} \tag{73}$$

As $A_1^* = (1, 1, 0) \notin C + W_1$ no output feedback decoupling is possible.

Example 3: (Falb and Wolovich [1].)

Let

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 0 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 1 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{74}$$

Then

$$D = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{75}$$

$$A^* = \begin{pmatrix} 1 & 2 & -1 \\ 0 & 1 & 0 \end{pmatrix} \tag{76}$$

As the system is not controllable, Theorem 3 is employed to determine the possibility of output feedback decoupling. It can be shown, using Lemma 4, that

$$W_1^c = \text{sp}\,\{(1, 0)\}, \; W_2^c = \text{sp}\,\{(0, 1)\} \tag{77}$$

As

$$\hat{A} = (A^* T_1^{-1})_{n_c} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \tag{78}$$

$$C^c = (C\,T_1^{-1})_{n_c} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{79}$$

$$-\hat{A}_1 = -C_1^c - C_2^c, \; -\hat{A}_2 = -C_2^c \tag{80}$$

it is easy to see that

$$H = \begin{pmatrix} 0 & \theta_{21} - 1 \\ \theta_{11} - 1 & -1 \end{pmatrix} \tag{81}$$

where θ_{21} and θ_{11} are arbitrary numbers.

The transfer matrix $T_d(s)$ of the closed-loop system is then given by

$$T_d(s) = \begin{pmatrix} \dfrac{\lambda_1}{s - \theta_{11} - 2} & 0 \\ 0 & \dfrac{\lambda_2}{s - \theta_{21} - 2} \end{pmatrix} \qquad (82)$$

As the open-loop transfer matrix $T(s)$ is

$$T(s) = \frac{1}{d(s)} \begin{pmatrix} s-1 & (s-1)^2 \\ (s-1)^2 & 0 \end{pmatrix} \qquad (83)$$

where $d(s) = (s-1)^3$

$$D\,T(s)^{-1} = \begin{pmatrix} s-1 & -1 \\ 0 & s-1 \end{pmatrix} \qquad (84)$$

Hence the system can be decoupled through output feedback. The diagonal elements of the closed-loop transfer matrix are given by

$$\left(T_d\right)_{11} = \lambda_1 \left[\left(DT(s)^{-1}\right)_{11} - \hat{h}_{11}\right]^{-1} = \lambda_1\left(s - \hat{h}_{11} - 1\right)^{-1} \qquad (85)$$

$$\left(T_d\right)_{22} = \lambda_2 \left[\left(DT(s)^{-1}\right)_{22} - \hat{h}_{22}\right]^{-1} = \lambda_2\left(s - \hat{h}_{22} - 1\right)^{-1}$$

where $\hat{h}_{11}$ and $\hat{h}_{22}$ are arbitrary numbers. Thus the Wolovich's method yields the same results.

V. CONCLUSIONS

A new necessary and sufficient condition for output feedback decoupling is obtained in this paper. As a consequence of this result, an easily computable condition is established which is expressed in terms of open loop parameters of the system.

ACKNOWLEDGEMENT

The Authors wish to express their thanks to Professor Elmer G. Gilbert of the University of Michigan and Professor William A. Wolovich of Brown University for their valuable suggestions and comments.

REFERENCES

[1] P. L. Falb and W. A. Wolovich, "Decoupling in the design and synthesis of multivariable control system," IEEE Trans. Automat. Contr., Vol. AC-12, pp. 651-659, 1967.

[2] E. G. Gilbert, "The decoupling of multivariable systems by state feedback," SIAM J. Contr. Vol. 7, No. 1, pp. 50-61, 1969.

[3] W. M. Wonham and A. S. Morse, "Decoupling and pole assignment in linear multivariable systems: a geometric approach," SIAM J. Contr. Vol. 8. No. 1, pp. 1-18, 1970.

[4] J. W. Howze, "Necessary and sufficient conditions for decoupling using output feedback," IEEE Trans. Automat. Contr., Vol. AC-18, pp. 44-46, 1973.

[5] W. A. Wolovich, "Output feedback decoupling," IEEE Trans. Automat. Contr., Vol. AC-20, pp. 148-149, 1975.

[6] S.-H. Wang and E. J. Davison, "Design of decoupled control systems: A frequency domain approach," Int. J. Contr., Vol. 21, No. 4, pp. 529-536, 1975.

[7] P. K. Sinha, "New condition for output feedback decoupling of multivariable systems," IEEE Trans. Automat. Contr. Vol. AC-24, pp. 476 - 478, 1979.

ANALYSIS OF SINGULAR SINGULARLY-PERTURBED SYSTEMS*

K. DAVID YOUNG
Department of Mechanical Engineering and Mechanics
Drexel University
Philadelphia, Pennsylvania 19104

ABSTRACT

Using singular perturbation techniques, a singular singularly-perturbed (SSP) system is decomposed into slow and fast subsystems. It is shown that SSP system response may contain high frequency oscillation. It is found that instead of the two time scales which exists in standard singularly-perturbed systems, multiple time scales may exist in SSP systems. Geometric properties of and the relationships among the system submatrices in SSP system dictate the multiple time scale behavior.

INTRODUCTION

In this paper we analyze the behavior of a class of linear singularly perturbed systems given by

$$\begin{aligned} \dot{x}_1 &= A_{11}x_1 + A_{12}x_2 \quad , \quad x_1 \varepsilon R^{n-m} \\ \mu\dot{x}_2 &= A_{21}x_1 + A_{22}x_2 \quad , \quad x_2 \varepsilon R^{m} \end{aligned} \tag{1a}$$

where A_{22} is a singular matrix. System (1a) is referred to as a singular singularly perturbed (SSP) system because the matrix A_{22} which is required to have nonzero eigenvalues with finite negative real parts in standard singularly perturbed systems is singular. This type of systems has been studied by O'Malley [1], Campbell and Rose [2] and Vasileva [3]. In [1], it is assumed that the eigenvalues of A_{22} are either simple zero eigenvalues or eigenvalues with negative real parts. A similar assumption is made in [2] and [3]. The work of Chow, Allemong and Kokotovic [4] dealt with the behavior of singularly perturbed systems that exhibit sustained high frequency oscillation. In [4], a system of the same form as (1) is considered with m being an even integer and the matrix A_{22} takes the special form

$$A_{22} = \begin{bmatrix} \mu A_1 & A_2 \\ A_3 & \mu A_4 \end{bmatrix} \tag{1b}$$

where A_2, A_3 are nonsingular matrices and the matrix product A_2A_3 has simple and negative eigenvalues. Although the matrix A_{22} in (1b) is nonsingular, the imaginary parts of $\lambda(\frac{A_{22}}{\mu})$ approach infinity as μ tends to zero, whereas the real parts tend to finite values. The work of Chow, et al. has extended the applications of singular perturbation methods to systems with sustained fast oscillation modes which until that time can only be applied to systems with the fast modes decaying exponentically in the fast time scale. We show, herein, that sustained fast oscillation modes can also appear in SSP systems. First, the totally SSP systems are considered and then we analyze SSP systems with the results obtained for totally SSP systems.

---

* This work is supported by the National Science Foundation under Grant No. ENG 79-21164.

Totally SSP System

When the matrix A_{22} in (1) is a null matrix, it is natural to refer to this special class of systems as totally SSP systems, that is,

$$\begin{aligned} \dot{x}_1 &= A_{11}x_1 + A_{12}x_2 \quad , \; x_1 \varepsilon R^{n-m} \\ \mu\dot{x}_2 &= A_{12}x_1 \qquad\qquad\;\; , \; x_2 \varepsilon R^{m} \end{aligned} \tag{2}$$

In contrast with the standard singularly perturbed systems, we show in this section that the decomposition of system (2) into slow and fast subsystems depends on the relative dimensions of x_1 and x_2. We shall deal with the two possibilities: (i) $n-m > m$ and (ii) $n-m \leqslant m$.

(i) $n-m > m$ case: We assume that the matrix A_{12} and A_{22} are of maximal rank and that the matrix product $A_{21}A_{12}$ is invertible. Then the decomposition of the state x_1 given by

$$x_1 = N_1 z_1 + A_{12}(A_{21}A_{12})^{-1} w_1 \tag{3}$$

is possible. The matrix N_1 is $(n-m) \times (n-2m)$ and its columns span $N(A_{21})$:

$$A_{21}N_1 = 0 \tag{4}$$

z_1 is an $(n-2m)$-vector representing the component of x_1 in $N(A_{21})$ and is defined by the expression

$$z_1 = M_1 x_1 \tag{5}$$

where the matrix M_1 is $(n-2m) \times (n-m)$ and its rows span $N(A_{12}^T)$. The m-vector w_1 represents the component of x_1 in $R(A_{12})$,

$$w_1 = A_{12}x_1. \tag{6}$$

The transformation defined by (3), (5) and (6) when applied to (2) yields

$$\dot{z}_1 = M_1A_{11}N_1z_1 + [M_1A_{11}A_{12}(A_{21}A_{12})^{-1} \quad 0] \begin{bmatrix} w_1 \\ x_2 \end{bmatrix} \tag{7}$$

$$\mu \begin{bmatrix} \dot{w}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu A_{21}A_{11}N_1 \\ 0 \end{bmatrix} z_1 + \begin{bmatrix} \mu A_{21}A_{11}A_{12}(A_{21}A_{12})^{-1} & \mu A_{21}A_{12} \\ I & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ x_2 \end{bmatrix} \tag{8}$$

Note that the dimension of the (w_1,x_2) - subsystem is 2m. The system (7), (8) is an SSP system in the form of (1). The corresponding matrix A_{22} is the matrix

$$D \equiv \begin{bmatrix} \mu A_{21}A_{11}A_{12}(A_{21}A_{12})^{-1} & \mu A_{21}A_{12} \\ I & 0 \end{bmatrix} \tag{9}$$

and clearly the rank of the 2mx2m matrix D is m as μ tends to zero. Since the system (7), (8) is obtained through a redefinition of the state variables of the totally SSP system (2), we have shown that it is premature to call (2) a totally SSP system. In view of this, we shall not make any further distinction between SSP and totally SSP systems.

We begin the analysis of the behavior of the system (7), (8) by examining the eigenvalues of $\frac{D}{\mu}$. The characteristic polynomial of $\frac{D}{\mu}$ is

$$\begin{aligned} \Delta(\lambda) &= \det\begin{bmatrix} \lambda I - \mu A_{21}A_{11}A_{12}(A_{21}A_{12})^{-1} & -\mu A_{21}A_{12} \\ - I & \lambda I \end{bmatrix} \\ &= -(-1)^m \det\left[\lambda^2 I - \lambda(A_{21}A_{11}A_{12}(A_{21}A_{12})^{-1}) - \frac{A_{21}A_{12}}{\mu}\right] \end{aligned} \tag{10}$$

Note that in the singular perturbation analysis of systems with sustained high frequency oscillation in [4], a characteristic polynomial similar to (10) is obtained in an attempt to expose the sustained high frequency oscillation. From Lemma 1 of [4], we obtain an asymptotic approximation of $\lambda(\frac{D}{\mu})$. Let T diagonalize $A_{21}A_{12}$ such that

$$TA_{21}A_{12}T^{-1} = \mathrm{diag}(-\omega_1, -\omega_2, -\omega_3, \ldots, -\omega_m) \tag{11}$$

and let the i^th diagonal element of $TA_{21}A_{11}A_{12}(A_{21}A_{12})^{-1}T^{-1}$ be donoted by σ_i.

Lemma 1: If the matrix $A_{21}A_{12}$ has simple and negative real eigenvalues $-\omega_i$, i=1, ..., m, then as $\mu \to 0^+$, the eigenvalues of $\frac{D}{\mu}$ approach infinity as

$$\sigma_i \pm j\sqrt{\omega_i}/\sqrt{\mu} \quad , \; i=1, \ldots, m. \tag{12}$$

Following the analysis in [4], we conclude that if the matrix $A_{21}A_{12}$ satisfy the assumption of Lemma 1, the system (7), (8) can be decomposed into a slowly varying subsystem and a fast oscillating subsystem. The following theorem summarizes the behavior of (2) when n-m > m.

Theorem 1: If n-m > m and the matrix $A_{21}A_{12}$ is nonsingular and its eigenvalues are simple negative real, then there exists a finite $T(\sqrt{\mu})$ such that for $0 \leq t \leq T$ the states of (2) are approximated to $0(\sqrt{\mu})$ by the slowly varying subsystem

$$\dot{\bar{z}}_1 = M_1A_{11}N_1\bar{z}_1 \quad , \quad \bar{z}_1(0) = z_1(0) \tag{13a}$$

$$\bar{x}_2(t) = -(A_{21}A_{12})^2 N_1 \bar{z}_1(t) \tag{13b}$$

and the oscillatory subsystem

$$\mu \begin{bmatrix} \dot{\tilde{w}}_1 \\ \dot{\tilde{x}}_2 \end{bmatrix} = D \begin{bmatrix} \tilde{w}_1 \\ \tilde{x}_2 \end{bmatrix} \tag{14a}$$

$$\tilde{w}_1(0) = w_1(0) \tag{14b}$$

$$\tilde{x}_2(0) = x_2(0) + (A_{21}A_{12})^2 N_1 z_1(0) \tag{14c}$$

as follows

$$z_1(t) = \bar{z}_1(t) + 0(\sqrt{\mu}) \tag{15}$$

$$\begin{bmatrix} w_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{x}_2(t) \end{bmatrix} + \begin{bmatrix} \tilde{w}_1(t) \\ \tilde{x}_2(t) \end{bmatrix} + 0(\sqrt{\mu}) \tag{16}$$

Moreover, if $M_1A_{11}N_1$ is Hurwitz and σ_i, $i=1, \ldots, m$ are negative then there exists a $\mu^* > 0$ such that the system (7), (8) is asymptotically stable for all $0 < \mu \leq \mu^*$.

Using (3) and the results in Theorem 1, the motion $x_1(t)$ can be analyzed. By substitution, we have

$$x_1(t) = N_1\bar{z}_1(t) + A_{12}(A_{21}A_{12})^{-1}\tilde{w}_1(t) + 0(\sqrt{\mu}). \tag{17}$$

In view of (13) and (14), we define the slowly varying part of x_1 as $\bar{x}_1$ and

$$\bar{x}_1(t) = N_1\bar{z}_1(t) + 0(\sqrt{\mu}). \tag{18}$$

This analysis justifies a simple method to obtain $\bar{x}_1(t)$, which is to set $\mu = 0$ in (2), as is usually done in singular perturbations. Consider an "output" of the x_1 - subsystem of (2)

$$y_1 = A_{21}x_1 \tag{19}$$

and an "input"

$$u_1 = x_2 \tag{20}$$

then setting $\mu = 0$ in (2) is equivalent to constraining the system

$$\dot{x}_1 = A_{11}x_1 + A_{12}x_2 \tag{21}$$

on the "zero output" subspace $y_1 = A_{21}x_1 = 0$. The dynamics of (21) in $y_1 = 0$ as shown in [5] is governed by an (n-2m) - dimensional system,

$$\dot{\sigma} = Z\sigma \tag{22}$$

where $\lambda(Z)$ are the transmission zeros of the triple (A_{21}, A_{11}, A_{12}). The motion $x_1(t)$ relates to $\sigma(t)$ through

$$x_1(t) = N_1 \sigma(t) \tag{23}$$

with N_1 defined in (4). Since $\lambda(Z) = \lambda(M_1 A_{11} N_1)$ the foregoing discussions illustrate a simple method to derive the slowly varying part of (2) which is justified by Theorem 1. We must caution, however, that this simple method does not reveal the fast oscillation part in (17).

If the nonsingular matrix $A_{21}A_{12}$ has simple positive real eigenvalues, the the matrix $\frac{D}{\mu}$ will have positive real eigenvalues of $0(\frac{1}{\sqrt{\mu}})$. The next lemma deals with the approximation of $\lambda(\frac{D}{\mu})$ when $A_{21}A_{12}$ has double real or a pair of complex eigenvalues.

Lemma 2: Suppose the matrix $A_{21}A_{12}$ has a pair of eigenvalues given by

$$\lambda_1, \lambda_2 = \gamma e^{\pm j\theta} \quad , \quad 0 \leq \theta < \pi, \ \gamma > 0 \tag{24}$$

then two pairs of the eigenvalues of $\frac{D}{\mu}$ tend to infinity as $\mu \to 0^+$ as

$$\frac{\gamma}{\sqrt{\mu}} e^{j(2k\pi-\theta)/2} \quad , \quad k = 0, 1 . \tag{25}$$

This lemma shows that for the fast subsystem of the system (2) to be asymptotically stable, the eigenvalues of $A_{21}A_{12}$ must neither be complex, nor double positive real. It can be shown that any eigenvalue of $A_{21}A_{12}$ with multiplicity greater than two would cause a right half plane eigenvalues of $0(\mu^{-1/k})$, $k > 2$ to appear in the fast subsystem.

(ii) $m \geq n-m$ case: Similar to the previous case, we assume that the matrix A_{12} and A_{22} are of maximal rank and that the matrix product $A_{12}A_{21}$ is invertible. Instead of decomposing x_1, the state x_2 is decomposed similar to (3)

$$x_2 = N_2 z_2 + A_{21}(A_{12}A_{21})^{-1} w_2 \tag{26}$$

The matrix N_2 is mx(2m-n) and its columns span $N(A_{12})$:

$$A_{12}N_2 = 0 \ , \tag{27}$$

z_2 is an (2m-n) vector ,

$$z_2 = M_2 x_2 \tag{28}$$

where the matrix M_2 is (2m-n) and its rows span $N(A_{21}^T)$. The (n-m)-vector w_2 is defined by

$$w_2 = A_{12}x_2. \tag{29}$$

The transformation defined by (26) (28) and (29) transforms (2) into decoupled subsystems,

$$\dot{Z}_2 = 0 \tag{30a}$$

and

$$\mu \begin{bmatrix} \dot{x}_1 \\ \dot{w}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} \mu A_{11} & \mu I \\ A_{12}A_{21} & 0 \end{bmatrix}}_{D'} \begin{bmatrix} x_1 \\ w_2 \end{bmatrix} . \tag{30b}$$

Consequently we conclude that under the assumed invertibility of $A_{12}A_{21}$, system (2) has, Viz. (30a), 2m-n simple zero eigenvalues. The subsystem (30b) is a fast oscillating system of the same type as (14a). The eigenvalues of (30b) are the roots of the characteristic polynomial

$$\Delta(\lambda) = -(-1)^{n-m}\det\,[\lambda^2 I - \lambda A_{11} - \frac{A_{12}A_{21}}{\mu}], \tag{31}$$

which is similar to (10). Thus the behavior of the fast subsystem (30b) will be similar to that of the fast subsystem in Theorem 1. Lemmas 1 and 2 are applicable if $A_{21}A_{12}$ is replaced by $A_{12}A_{21}$. There is no need to approximate the response of (2) as it is done in Theorem 1 since (30a) is the slowly varying subsystem and (30b) is the fast oscillating subsystem. The motions $x_1(t)$ and $x_2(t)$ are given by

$$x_2(t) = N_2M_2x_2(0) + A_{21}(A_{12}A_{21})^{-1}w_2(t), \tag{32}$$

$$\begin{bmatrix} x_1(t) \\ w_2(t) \end{bmatrix} = e^{D'\frac{t}{\mu}} \begin{bmatrix} x_1(0) \\ A_{12}x_2(0) \end{bmatrix} . \tag{33}$$

SSP Systems

Consider the SSP System (1) when A_{22} has ℓ nonzero and m-ℓ simple zero eigenvalues. In this case it is possible to factor A_{22} such that

$$A_{22} = B_2C_2 \tag{33}$$

where B_2 is m×ℓ, C_2 is ℓ×m and C_2B_2 is invertible. The following decomposition conveniently reveals the fast and slow behavior of (1),

$$X_2 = N_2\, z_2 + B_2(C_2B_2)^{-1} w_2 \tag{34}$$

where N_2 is mx(m-ℓ), $\mathcal{R}(N_2)$ spans $\mathcal{N}(C_2)$,

$$z_2 = M_2x_2, \tag{35}$$

M_2 is (m-ℓ) xℓ, $\mathcal{R}(M_2^T)$ spans $\mathcal{N}(B_2^T)$ and

$$w_2 = C_2\, x_2. \tag{36}$$

System (2) takes the form

$$\begin{bmatrix} \dot{x}_1 \\ \mu\,\dot{z}_2 \\ \mu\,\dot{w}_2 \end{bmatrix} = \left[\begin{array}{cc|c} A_{11} & A_{12}N_2 & 0 \\ M_2A_{21} & 0 & 0 \\ \hline C_2A_{21} & 0 & C_2B_2 \end{array}\right] \begin{bmatrix} x_1 \\ z_2 \\ w_2 \end{bmatrix} \tag{37}$$

The eigenvalues of (37) are clearly the eigenvalues of the main diagonal blocks. Since $\lambda(C_2B_2)$ are the nonzero eigenvalues of A_{22}, stability in the $\frac{t}{\mu}$ time scale is guaranteed by $\mathcal{R}e\lambda(A_{22}) \leq 0$. The upper diagonal block matrix has similar structure as a totally SSP system. Thus the results in the previous section are applicable: if $n-m > m$ then case (i) applies and if $m > n-m$, then case (ii) applies. We shall discuss here only the case when $n-m > m$. For the results of case (i) to apply, we need $M_2A_{21}A_{12}N_2$ to be nonsingular. We note that nonsingularity of $A_{21}A_{12}$ as we have assumed before is not sufficient to grarantee the invertibility of $M_2A_{21}A_{12}N_2$. However these are two cases for which invertibility of $A_{21}A_{12}$ is a sufficient condition. these cases are: (i) $A_{22}A_{21}A_{12}$ has the same null space as A_{22} and (ii) $A_{21}A_{12}A_{22}$ has the same range space as A_{22}. Under these conditions, the analysis of system (37) can proceed along the line of separating the slow and fast oscillating subsystems as given in Theorem 1.

Conclusions

It is found that the decomposition of the state space into fast and slow subspaces changes drastically as the relationship between the dimensions of x_1 and x_2 is the SSP system (1a) change from (i) $n-m \geq m$ to (ii) $n-m < m$. This is different from the fast and slow decomposition of standard singularly -perturbed systems which is invariant with respect to the dimensions of x_1 and x_2. The singularity of A_{22} requires a much closer examination of the geometric properties of and relationships among the system submatrices, in particular, A_{22},A_{21}, and A_{12}. We have shown that generally a multiple time scale behavior instead of a two time scale behavior prevails in SSP systems. We have discussed in this paper the behavior of a selected class of SSP system structures. Due to space restriction, results on other SSP structures will be discussed elsewhere.

References

[1] O'Malley, R.E., Jr., "A Singular Singularly-Perturbed Linear Boundary Value Problem" SIAM J. Math. Anal., Vol. 10, pp. 695-708, 1979.

[2] Campbell, S.L. and N.J. Rose, "Singular Perturbation of Autonomous Linear System," SIAM J. Math. Anal., Vol. 10, pp. 542-551, 1979

[3] Vasileva, A.B., "Singularly Perturbed Systems Containing Indeterminacy in the Case of Degeneracy," Soviet Math. Dokl., Vol. 16, No. 5, pp. 1121-1125, 1975.

[4] Chow, J.H., J.J. Allemong and P.V. Kokotovic, "Singular Perturbation Analysis of systems with sustained high frequency oscillations" Automatica, Vol. 14, pp-271-279, 1978.

[5] Young, K-K.D., P.V. Kokotovic and V.I. Utkin, "A Singular Perturbation Analysis of High gain feedback systems," IEEE Trans. Auto. Contr., Vol. Ac-22, pp. 931-938, 1977.

DIGITALIZATION OF THE CONTINUOUS-DATA SYSTEMS VIA TIME-RESPONSE MATCHING

CHYAN-GOEI CHUNG, KWANG-WEI HAN
Chiao Tung University
Hsinchu, Taiwan
Republic of China

HSI-HAN YEH
Department of Electrical Engineering
University of Kentucky
Lexington, Ky 40506

Abstract

A method for the design of digital controllers by matching the output time responses of digital control systems with those of continuous models is presented in this paper. The optimum coefficients of digital controllers are found by using Golub's algorithm to solve the least square-errors problem. The advantages of this method are: (1) the order of digital controller is low, (2) the required sampling frequency is low, (3) the accuracy is high, and (4) the response between sampling instants can be matched. An example is given and comparisons with the results in current literature are made.

I. Introduction

In recent years, most of the control systems are implemented by use of digital controllers instead of continuous compensators. Usually an on-line digital computer must control several loops simultaneously, so the sampling period should be chosen as large as possible.

Many works [1]-[5] have been done for the design of digital controllers by matching the time responses of digital control systems with those of continuous models. But these methods only consider the responses at sampling instants, the results of design may be unsatisfactory in the intersampling periods.

In this paper, a computer-aided method for the design of digital controllers via time response matching is proposed. The output responses of digital control systems and models are calculated by the fourth-order Runge-Kutta method. A large number of samples are chosen at the sampling instants and between the sampling instants. Golub's algorithm for solving least squares problems is used iteratively to find the optimum coefficients of digital controllers.

II. Review of Golub's Algorithm for Least Square-Error Problems

Golub's algorithm [6] is a numerical algorithm for solving least squares problems in a high accurate manner. Breen et al. [7] and Kao [8] have applied this algorithm in circuit optimization successfully. The basic concept is described briefly as follows.

Consider the system expressed by

$$\underline{A}\,\underline{X} = \underline{B} \tag{1}$$

where $\underline{A}$ is a given nxm real matrix of rank m, $\underline{B}$ is a given nx1 real column vector, and $\underline{X}$ is an mx1 column vector. Eq. (1) has a unique solution if n=m. Suppose n>m, and we wish to find a vector $\hat{\underline{X}}$ such that

$$||\underline{B} - A\,\hat{\underline{X}}|| = \min \tag{2}$$

where $||....||$ represents the euclidean norm. Choosing an orthogonal matrix $\underline{Q}$ such that

$$\underline{Q}\,\underline{A} = \underline{R} = \begin{bmatrix} \tilde{\underline{R}} \\ \text{---} \\ \underline{0} \end{bmatrix} \tag{3}$$

where $\underline{0}$ is an (n-m)xm zero matrix and $\tilde{\underline{R}}$ is an mxm upper triangular matrix as

$$\underline{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ 0 & r_{22} & \cdots & r_{2m} \\ & & \ddots & \vdots \\ 0 & & & r_{mm} \end{bmatrix} \tag{4}$$

we have

$$||\underline{B} - \underline{A}\ \underline{X}|| = ||\underline{Q}\ \underline{B} - \underline{Q}\ \underline{A}\ \underline{X}|| = ||\underline{C} - \underline{R}\ \underline{X}|| \tag{5}$$

where $\underline{C}=\underline{Q}\ \underline{B}$. Eq. (5) can be rewritten as

$$||\underline{B}-\underline{A}\ \underline{X}|| = [(c_1-r_{11}x_1-r_{12}x_2-\ldots.-r_{1m}x_m)^2+(c_2-r_{22}x_2-r_{23}x_3-\ldots.-r_{2m}x_m)^2 + \ldots. +(c_m-r_{mm}x_m)^2+c_{m+1}^2+c_{m+2}^2+\ldots.+c_n^2]^{1/2} \tag{6}$$

Thus $||\underline{B}-\underline{A}\ \underline{X}||$ is minimized when

$$\left.\begin{aligned} r_{11}x_1+r_{12}x_2+ \ldots. +r_{1m}x_m &= c_1 \\ r_{22}x_2+ \ldots. +r_{2m}x_m &= c_2 \\ \ddots \qquad \vdots \quad & \\ r_{mm}x_m &= c_m \end{aligned}\right\} \tag{7}$$

and the minimized value is

$$||\underline{B} - \underline{A}\ \underline{X}|| = (c_{m+1}^2+c_{m+2}^2+\ldots.+c_n^2)^{1/2} \tag{8}$$

The decomposition of Eq. (3) can be realized by Householder transformation [6],[8].

For least square-errors problems, assume that the cost function F is defined as

$$F = \sum_{i=1}^{n} e_i^2(\underline{d},t_i) \tag{9}$$

where t_i represents the independent variable in the time range of interest, $\underline{d}$ is a dependent variable vector with m elements, e_i is the error function at t_i, and n is the total number of sampled points in the time range of interest. The purpose is to find the optimum parameter vector $\underline{d}$ such that the cost function F is minimized. The partial derivative of $e_i(\underline{d})$ with respect to $\underline{d}$ is $\underline{e}_i'(\underline{d})=(\partial e_i(\underline{d}))/(\partial \underline{d})$, which is an m-dimensional row vector. The Taylor's series of $e_i(\underline{d})$ is

$$e_i(\underline{d}+\Delta\underline{d})=e_i(\underline{d})+\underline{e}'(\underline{d})\Delta\underline{d}+\frac{1}{2!}\underline{e}_i''(\underline{d})(\Delta\underline{d})^2+\ldots. \tag{10}$$

If $\Delta\underline{d}$ is very small, the second order and higher order terms can be neglected, then we get

$$e_i(\underline{d}+\Delta\underline{d}) \cong e_i(\underline{d})+\underline{e}_i'(\underline{d})\Delta\underline{d} \tag{11}$$

In order to minimized the cost function $F(\underline{d})$, the direction vector $\Delta\underline{d}$ should be in the direction so that $e_i(\underline{d}+\Delta\underline{d})=0$. Therefore in view of Eq. (11) $\Delta\underline{d}$ can be determined by the equation

$$e_i(\underline{d}) + \underline{e}_i'(\underline{d})\Delta\underline{d} = 0 \qquad i=1,2,\ldots.,n \tag{12}$$

or

$$\underline{e}_i'(\underline{d})\Delta\underline{d} = -e_i(\underline{d}) \tag{13}$$

which can be rewritten as

$$\begin{pmatrix} \frac{\partial e_1}{\partial d_1} & \frac{\partial e_1}{\partial d_2} & \cdots & \frac{\partial e_1}{\partial d_m} \\ \frac{\partial e_2}{\partial d_1} & \frac{\partial e_2}{\partial d_2} & \cdots & \frac{\partial e_2}{\partial d_m} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_n}{\partial d_1} & \frac{\partial e_n}{\partial d_2} & \cdots & \frac{\partial e_n}{\partial d_m} \end{pmatrix} \begin{pmatrix} \Delta d_1 \\ \Delta d_2 \\ \cdot \\ \cdot \\ \Delta d_m \end{pmatrix} = \begin{pmatrix} -e_1 \\ -e_2 \\ \cdot \\ \cdot \\ -e_n \end{pmatrix} \tag{14}$$

Since $n>m$, Golub's algorithm can be applied to solve Eq. (13) and to find the value of $\Delta\underline{d}$. But due to the approximation made in Eq. (11) and the fact that $\Delta\underline{d}$ is only a least squares solution, the new parameter vector $\underline{d}+\Delta\underline{d}$ does not, in general, minimize the cost function $F(\underline{d})$. The direction is only an indication that if $\underline{d}$ is changed infinitesimally in the direction of $\Delta\underline{d}$ then the cost function $F(\underline{d})$ will be reduced. This means

$$F(\underline{d} + \alpha\Delta\underline{d}) < F(\underline{d}) \tag{15}$$

where α is called the step size and $0<\alpha\le 1$. The optimum step size α can be found by the divide-by-8 algorithm [8].

Arbitrarily choose a $\underline{d}^o$ as the initial value of $\underline{d}$. Using above procedures to find $\Delta\underline{d}^o$ and α^o, we get $\underline{d}^1=\underline{d}^o+\alpha^o\Delta\underline{d}^o$ as the new value of $\underline{d}$, with $F(\underline{d}^o)>F(\underline{d}^1)$. Repeat the above procedures using $\underline{d}^{k+1}=\underline{d}^k+\alpha^k\Delta\underline{d}^k$, $k=0,1,2,\ldots$, until $F(\underline{d}^i+\alpha^i\Delta\underline{d}^i)$ or $F(\underline{d}^i)-F(\underline{d}^i+\alpha^i\Delta\underline{d}^i)$ is less than a tolerable value. Then $\underline{d}^{i+1}$ is the optimum parameter vector.

III. Statement of The Problem

Consider the continuous model as shown in Fig. 1. Now a digital controller and a zero-order hold are used to replace the continuous compensator as shown in Fig. 2. The general form of the transfer function of the digital controller can be written as

$$D(z) = \frac{a_o+a_1z^{-1}+\ldots+a_mz^{-m}}{1+b_1z^{-1}+\ldots+b_nz^{-n}} \tag{16}$$

where a's and b's are constants, and $\underline{a}=(a_o,a_1,\ldots,a_m)$ $\underline{b}=(b_1,b_2,\ldots,b_n)$. The problem is to find the unknown constants a's and b's so that the output response of the digital control system matches that of the continuous model as closely as possible. In other words, it is required to find the coefficient vectors $\underline{a}$ and $\underline{b}$ which will minimize the cost function

$$F(a,b) = \int_o^\infty [y(t,\underline{a},\underline{b})-c(t)]^2 dt \tag{17}$$

IV. The Design Method

Given $\underline{a}$ and $\underline{b}$, the fourth-order Runge-Kutta method and the shifting process for the numerical solution of difference equation can be applied to evaluate the output responses of the continuous model and the digital control system. Let the step size of time ΔT in Runge-Kutta method be chosen as $\Delta T=(1/k)T$, where T is the sampling period of the digital control system and k is a positive integer. The responses of both the continuous model and the digital control system to a unit-step input are evaluated at instants 0, ΔT, $2\Delta T$, ..., etc. Thus, the cost function of Eq. (17) can be discretized. Since ΔT is very small, the step size of the discretization can be taken as $\beta\Delta T$, for some positive integer β. Thus the cost function may be reduced to

$$F(\underline{a},\underline{b}) = \sum_{i=1}^{N} [y(t_i,\underline{a},\underline{b}) - c(t_i)]^2 \tag{18}$$

where $t_i = i \times \beta \times \Delta T$, and N is an integer such that t_N should be large enough to cover the transient response and a part of the steady state response. Note that β may be less than, equal to, or greater than k, and the value of β is dependent on the system considered and the sampling period chosen.

In Fig. 1, and 2, assume r(t), P(s), G(s), H(s), T, n and m are given. To initiate the application of Golub's Algorithm for the minimization of the cost function given in Eq. (18), initial values of $\underline{a}$ and $\underline{b}$ may be arbitraily set, provided that the resulting closed-loop system is stable. Let the initially guessed values of $\underline{a}$ and $\underline{b}$ be $\underline{a}^o$ and $\underline{b}^o$, respectively. Then by use of Runge-Kutta method and shifting process the output responses $y(t_i\ ,\underline{a}^o,\underline{b}^o)$ and $c(t_i)$ can be evaluated. Using the procedures described in Section II, the coefficient-correction vectors $\Delta\underline{a}^o$ and $\Delta\underline{b}^o$ and the optimum step size α^o can be obtained. The new coefficient $\underline{a}^1$ and $\underline{b}^1$ are

$$\begin{aligned} a^1 &= a^o + \alpha^o \Delta \underline{a}^o \\ \underline{b}^1 &= \underline{b}^o + \alpha^o \Delta \underline{b}^o \end{aligned} \tag{19}$$

with $F(\underline{a}^1,\underline{b}^1) < F(\underline{a}^o,\underline{b}^o)$. Repeat the above procedures iteratively until $F(\underline{a}^j,\underline{b}^j)$ or $F(\underline{a}^{j-1},\underline{b}^{j-1})-F(\underline{a}^j,\underline{b}^j)$ is equal to or less than a preassumed value, then $\underline{a}^j$ and $\underline{b}^j$ are the optimum coefficients of the digital controller.

If the desirable result can not be found, the order of the digital controller should be increased, and the design process is repeated again.

V. Numerical Example

In Fig 1, let H(s)=1,

$$G(s) = \frac{6.25(s+0.5)(s+0.1)}{(s+5)(s+0.01)} \quad \text{and} \quad P(s) = \frac{4}{s(s+0.5)}\ .$$

The closed-loop function is [5]

$$\frac{C(s)}{R(s)} = \frac{25s+2.5}{s^3+5.01s^2+25.05s+2.5} \tag{20}$$

Suppose r(t) is a unit-step function and we are interested in the magnitudes of the time response at the following instants:

$$t_i = i \times 0.1 \text{ sec} \qquad i=1,2,....,40 \tag{21}$$

For various values of T, the optimum 1st-order digital controllers are found as

$$D(z) = \frac{5.116894z-4.799533}{z-0.490811} \qquad (T=0.1 \text{ sec.}) \tag{22}$$

$$D(z) = \frac{4.127699z-3.642406}{z-0.029834} \qquad (T=0.2 \text{ sec.}) \tag{23}$$

$$D(z) = \frac{3.276658z-2.724734}{z+0.344619} \qquad (T=0.3 \text{ sec.}) \tag{24}$$

$$D(z) = \frac{2.487124z-1.962635}{z+0.577060} \qquad (T=0.4 \text{ sec.}) \qquad (25)$$

The results are plotted in Fig. 3. The digital controllers obtained by point-by-point output matching method are found as

$$D(z) = \frac{3.586(z-.8607)(z-.9962)(z+.5993)}{(z+0.9508)(z-0.9813)(z+0.1364)} \qquad (T=.3 \text{ sec.}) \qquad (26)$$

$$D(z) = \frac{2.931(z-0.8187)(z-0.9204)(z+0.4916)}{(z+0.9356)(z-0.9789)(z+0.2717)} \qquad (T=.4 \text{ sec.}) \qquad (27)$$

The results are plotted as shown in Figs. 4 and 5, where the superior characteristics of the presented method can be realized directly, especially the intersampling-period response as shown in Fig. 5.

VI. Conclusions

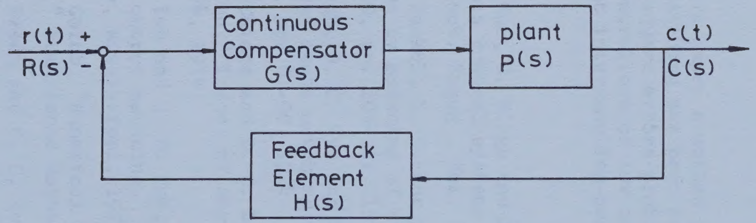
In this paper a method for the design of digital controllers via time response matching has been presented. By use of Golub's algorithm to solve the least square-errors problem the presented method can be used to design digital controllers of low order, high accuracy and low sampling frequency, and without intersampling-period oscillations.

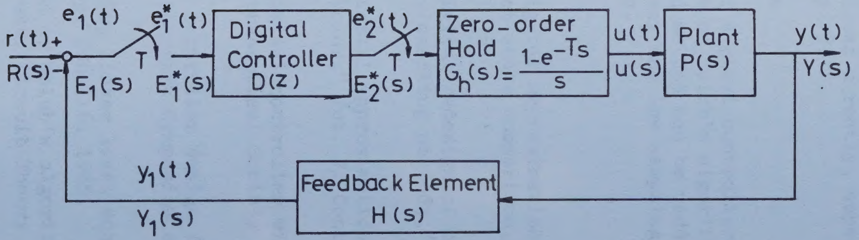
References

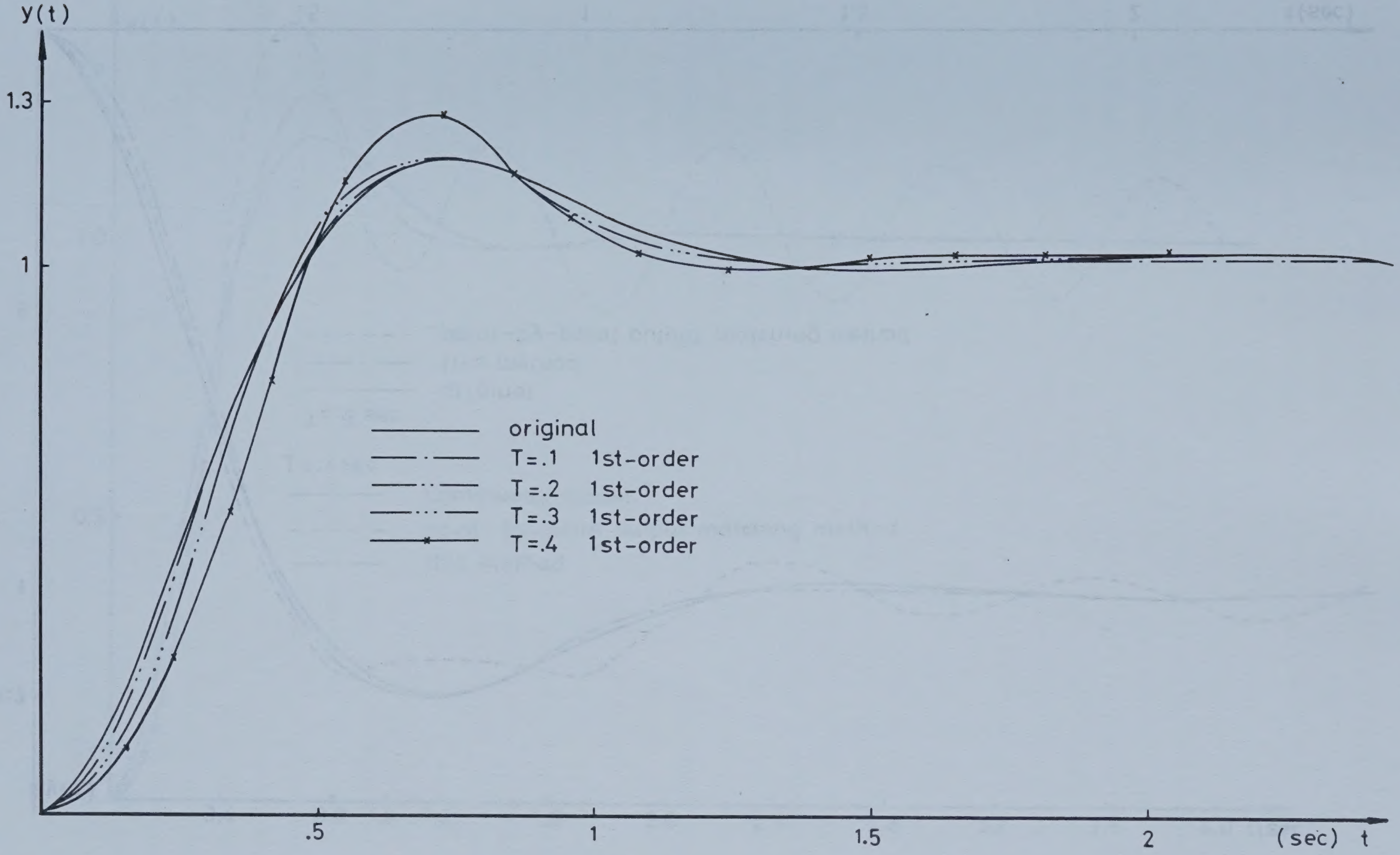
1. B. C. Kuo, G. Singh and R. A. Yackel, "Digital approximation of continuout-data control systems by point-by-point state comparison," Computer and Elect. Engng., Vol. 1, pp. 155-170, 1973.
2. R. A. Yackel, B. C. Kuo and G. Singh, "Digital redesign of continuous systems by matching of states at multiple sampling periods," Automatica, Vol. 10, pp. 105-111, 1974.
3. G. Singh, B. C. Kuo and R. A. Yackel, "Digital approximation by point-by-point state matching with high-order holds," Int. J. Control, Vol. 20, No. 1, pp. 81-90, 1974.
4. S. M. Chiang and D. C. Liu, "Design of digital controller via Walsh function," Trans. of Aeronautical and Astranautical Society of ROC, pp. 170-194, 1974.
5. C. S. Yeh and H. H. Yeh, "On the digital controller design for point-by-point output matching," 10th Annual Southeastern Symposium on System Theory, Mississippi, 1978.
6. G. H. Golub, "Numerical Methods for solving linear least squares problems," Numerische Mathematic, Vol. 7, pp. 206-216, 1965.
7. R. H. Breen and G. C. Temes, "Application of Golub's algorithm in circuit optimization and analysis," IEEE Trans. Circuit Theory, Vol. CT-20, pp. 687-690, 1973.
8. I. C. Kao, "Modification of Golub's algorithm for circuit optimization," Ph.D. dissertation, Southern Methodist University, 1975.

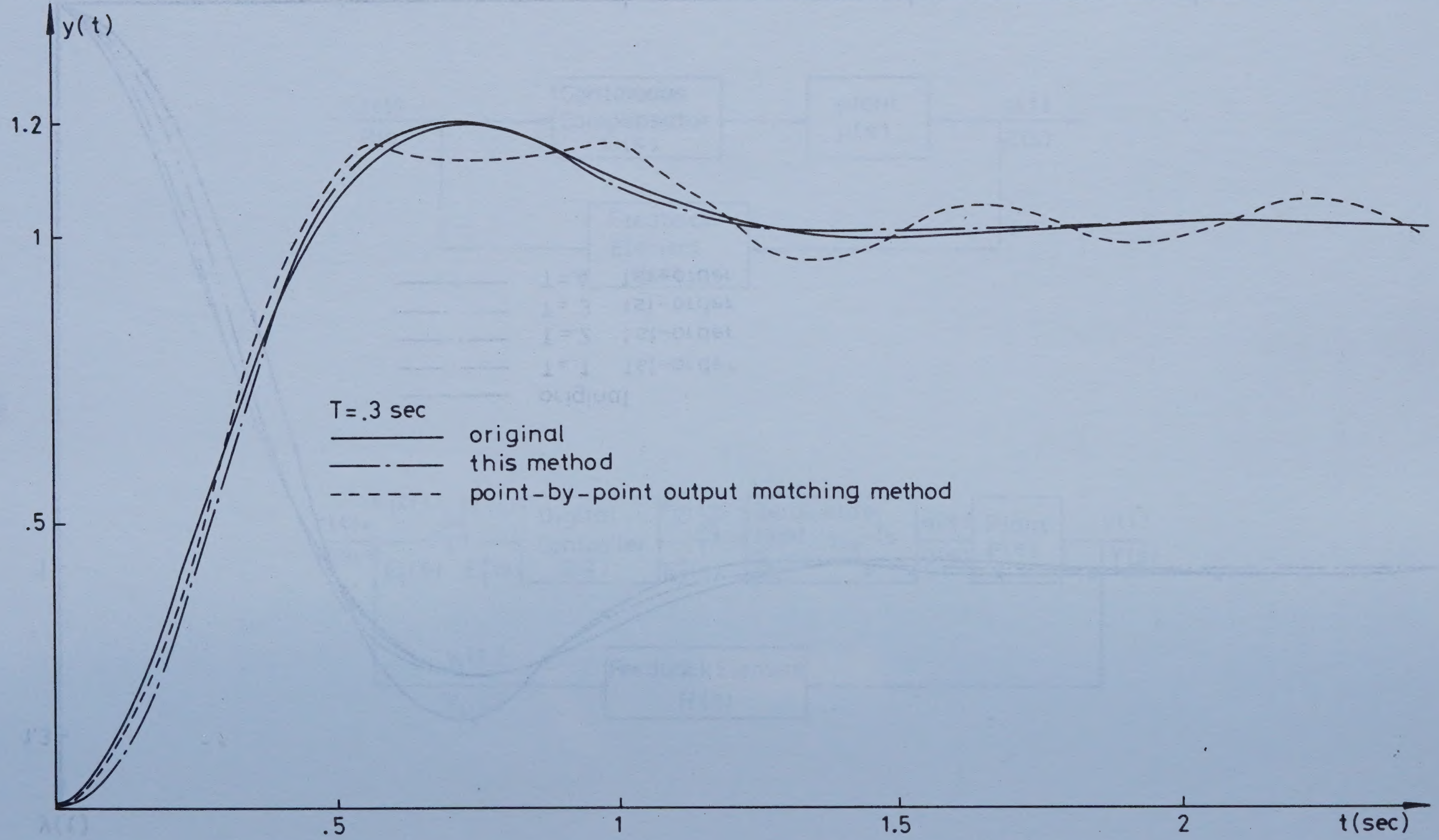
LIST OF CAPTIONS

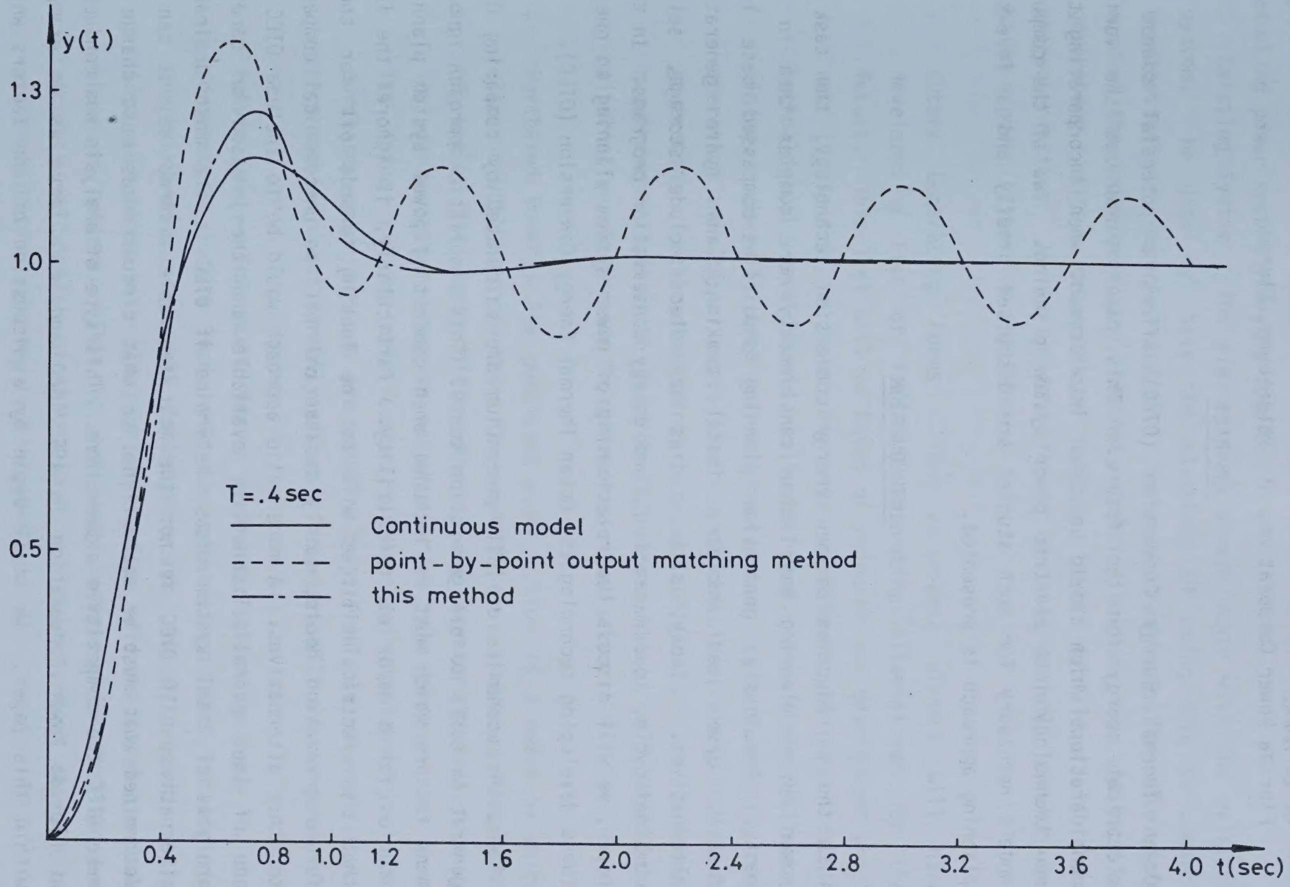
Fig. 1 Continuous model.
Fig. 2 Control system with a digital controller.
Fig. 3 Step responses for various values of T.
Fig. 4 Step responses for different digital controllers.
Fig. 5 Step responses for different digital controllers.


r(t) +
R(s) -
Continuous Compensator G(s)
plant P(s)
c(t)
C(s)
Feedback Element H(s)


r(t)+
R(s)-
e1(t)
E1(s)
T
e1*(t)
E1*(s)
Digital Controller D(z)
e2*(t)
E2*(s)
T
Zero-order Hold Gh(s) = (1-e^-Ts)/s
u(t)
u(s)
Plant P(s)
y(t)
Y(s)
Feedback Element H(s)
y1(t)
Y1(s)


y(t)
1.3
1
.5
1
1.5
2
(sec) t
original
T=.1 1st-order
T=.2 1st-order
T=.3 1st-order
T=.4 1st-order


y(t)
1.2
1
.5
.5
1
1.5
2
t(sec)
T=.3 sec
original
this method
point-by-point output matching method


y(t)
1.3
1.0
0.5
0.4
0.8
1.0
1.2
1.6
2.0
2.4
2.8
3.2
3.6
4.0
t(sec)
T=.4 sec
Continuous model
point-by-point output matching method
this method

INTEGRATING OTEC INTO POWER SYSTEM PLANNING STUDIES

G. E. MARKS
Florida Power Corporation, St. Petersburg, Florida

ABSTRACT

Ocean Thermal Energy Conversion (OTEC) offers a potential source of electrical energy for the future. This paper summarizes the various considerations which should be taken into account when incorporating this new technology into electric power system planning. Two of the computer models necessary for such studies are discussed briefly and a five-step planning approach is presented.

INTRODUCTION

With the introduction of new energy conversion technology, the task of power system planning has become considerably more complex than in the past. Previously, generation planning studies encompassed base load fossil, base load nuclear, fossil peaking, and hydro generation alternatives. Today, such studies may also include storage, solar, combined-cycle, load management, and energy conservation programs. In this paper, we will discuss the relationship of power system planning to one of these developing technologies, Ocean Thermal Energy Conversion (OTEC).

Often, the economics of OTEC generation are calculated by comparing OTEC generation costs to oil generation costs. This simplistic approach ignores many factors which must be included when conducting power system planning studies for a major electric utility. Particularly, it ignores the fact that the electric utilities will not be looking towards oil for their future generation needs, mainly because oil will be uneconomical compared to other alternatives. A more valid approach would be to include OTEC as one of the several alternatives available and then, based on careful analysis of total system costs, determine if OTEC is the most desirable alternative. If OTEC is not the most favorable alternative, it can be determined what must be accomplished or what circumstances must change to make OTEC the competitive alternative. This type of analysis was conducted at Florida Power Corporation in 1979/1980 and will serve as an example within this paper. We shall begin by a discussion of the factors which must be included in power system planning studies and their relationship to the OTEC alternative.

PLANNING CONSIDERATIONS

Figure 1 illustrates a number of factors which must be considered when conducting power system planning studies.

- Existing System - The planner must always begin with the existing system. He does not have the advantage of being able to conduct hypothetical or generic studies. He must realistically take into account the local geography, weather, existing mixture, age and type of generating units, transmission system, load profile, and load characteristics. It is on this foundation that he will develop and build a power system for the future.

- Future Generating Forms - The up-to-date planner will have developed a list of feasible generating alternatives for the future. This list will be based on research and development which has transpired to date and his assessment of what future technological developments will be made, and their estimated timing. He will need to acquire expert input concerning the operating and capital costs of these alternatives, along with their generating characteristics, availabilities, maintenance requirements, etc. It is from this list that he will choose the alternatives to include in the economic evaluation.

- Purchased Power - The purchased power option is a means by which electric utilities can share the benefits of unit timing, economies of scale, fuel cost differentials, and weather diversity. To include purchased power as an alternative, the planner must know its fixed and variable costs, timing, and availability. The purchased power options are usually only available in the short-term and would not normally affect a long-term OTEC evaluation.

- Future Load - The basis for a generation planning study is the forecast of future load. To be valuable, it must include the energy forecast, the peak demand forecast, and a forecast of the load variations with time, which is often referred to as the load shape or pattern. This is absolutely necessary for detailed and accurate calculation of the generation dispatch, fuel consumption and contribution from new technologies.

Regulatory Constraints - Working within a regulated industry, power system planners must recognize certain constraints in the development of plans or face severe penalties. For example, Public Service Commissions establish a maximum allowable rate of return for electric utilities. This is done in the public interest, since a utility constitutes a type of monopoly within its own service area. However, Commissions do not guarantee that the utility will earn such rate of return and often further constrain the utility to such a degree that even the permissible value is seldom achieved. Commissions also have the power to eliminate certain investments from the utilty rate base if they deem the decision or its implementation were not prudent. Hence, the utility runs the risk, for example, of having a plant being removed from the rate base as a result of a catastrophic failure or discontinued operation due to excessive operating costs or hazards. These constraints make it difficult, if not impossible, for an electric utility to enter into any type of high-risk generating venture.

Government Incentives - There may be circumstances which make certain alternatives desirable on a national basis. This could be due to such things as environmental objectives, defense considerations, or international economic forces. Such factors would not necessarily be evident in the economic evaluations of a local electric utility. However, such biases can be injected into local planning studies through the development of governmental incentives. By including such incentives in the evaluation, the local utility planner is essentially adjusting his results to include national considerations.

Financial Considerations - The planner must take into account a realistic assessment of the corporate financial situation, and the availability of funds on the national market. The electric utility industry is probably the most capital-intensive industry in the nation. A utility must watch its debt ratios carefully or swiftly erode its financial integrity. Furthermore, it is possible that with increasing construction costs, an electric utility might not be able to attract the capital necessary to implement a given construction program. This already critical problem becomes acute when integrating the developing solar technologies into todays power systems. Loan guarantees, unfortunately, do not solve the problem, since the utility is still liable for such debts. Loan guarantees are only beneficial if the utility were to go bankrupt. In the interim, they must be shown on the books as liabilities and, hence, impact the corporate debt and financial posture.

- Corporate Policies - Businesses are run based on corporate policy just as government is run based on national policy. It is the established corporate policy that distinguishes well-run from poorly-run electric utilities. All utilities should not have identical policies due to the unique characteristics of each system, corporation and community. Corporate policies include such things as: reliability guidelines, acceptable debt ratios, and the level of risk which can be assumed.

PLANNING MODELS

Taking into account both technical and institutional considerations, the planner must develop the most desirable overall generation, transmission and distribution expansion plan. Obviously, sophisticated tools will be required to model the technical aspects of power system design and operation. A basic question is, "Can these tools model OTEC considerations for power system planning purposes?" We found that existing tools could model OTEC satisfactorily for economic evaluation purposes. The following is a discussion of the two major tools needed for such an evaluation.

- Production Simulation - Production simulation is necessary for the determination of fuel and operating costs for each generation alternative. The alternative cannot be looked at in an isolated sense. It must be incorporated into the overall system with total costs calculated including this alternative as an integral part of that system. In such a way, the true overall economics can be assessed. At the present time, Florida Power Corporation is using the PROMOD production simulation program. Figure 2 illustrates the various inputs necessary for running this model and the resulting outputs. Using PROMOD or a similar production costing program, the hourly operation of an electric utility can be simulated with the proposed alternative fully integrated into the system. Planning studies can then be conducted representing 20 or more years of power system operation. Key outputs are the loss of load probability (LOLP) which is a measure of system reliability, the monthly energy and demand requirements for the system, and the monthly generation, fuel consumption, fuel costs, fixed and variable O&M costs for the system and for each generating unit. We found that by careful construction of the OTEC model, no changes to the production simulation program were required to represent OTEC operation for generation costing purposes. This, however, might not be true of other solar technologies.

Corporate Simulation - A corporate model has been developed which simulates the financial operation of the company. Figure 3 illustrates the input and output parameters for this complex model. Notice that the production simulation program results constitute one of these inputs. In addition, the capital budget including construction costs of the plan under consideration are input to the model. The sales forecast and rate structure are included to permit the model to develop a forecast of corporate revenues. The model at Florida Power Corporation represents 2500 variables and will produce income statements and balance sheets for the years under study. Of particular interest is the revenue analysis which is a key parameter for evaluation of the expansion plans under consideration. A calculation of the present worth of the cumulative total revenue requirements (CTRR) necessary to support each proposed expansion plan is essentially an estimate of the cost of that plan to the utility consumer over the study horizon. The goal of the electric utility planner is to select the plan which minimizes the total cost to the consumer while taking into account reliability, environmental and regulatory considerations.

Numerous other models are used in the design of the electric power transmission grid and distribution system. These will be omitted here due to space and time limitations.

THE PLANNING PROCESS

Given these tools and considerations, the planner is now faced with the challenge of conducting an organized and accurate evaluation of the various possible future system generation plans. Figure 4 illustrates the five-step approach which was used in the current OTEC evaluation. OTEC was compared to the most competitive plan presently available, base load coal fired steam plants. In this study, it was hypothesized that an OTEC plant would be available by the late 1980's. This time period was selected because it was the period currently being evaluated by the utility. Although it would not be feasible to construct a plant of the size represented (400 MW) by that timeframe, it did enable an evaluation of how OTEC would have fared in such an economic comparison had it been available. More important, however, an effort was made to determine what factors would need to change to improve OTEC competitiveness.

CONCLUSION

An economic evaluation of OTEC can be included in power system planning studies today. Existing computer programs can be used to model production simulation and financial requirements for economic comparisons involving the OTEC alternative. Care must be taken, however, to take into account institutional as well as technical factors and to calculate overall system costs rather than conducting fuel type or unit-by-unit comparisons.

This work was funded by the Department of Energy (DOE) under contract No. DE-AC02-79ET29187. The results and conclusions within this paper reflect the views of the author and are not necessarily the views of DOE.

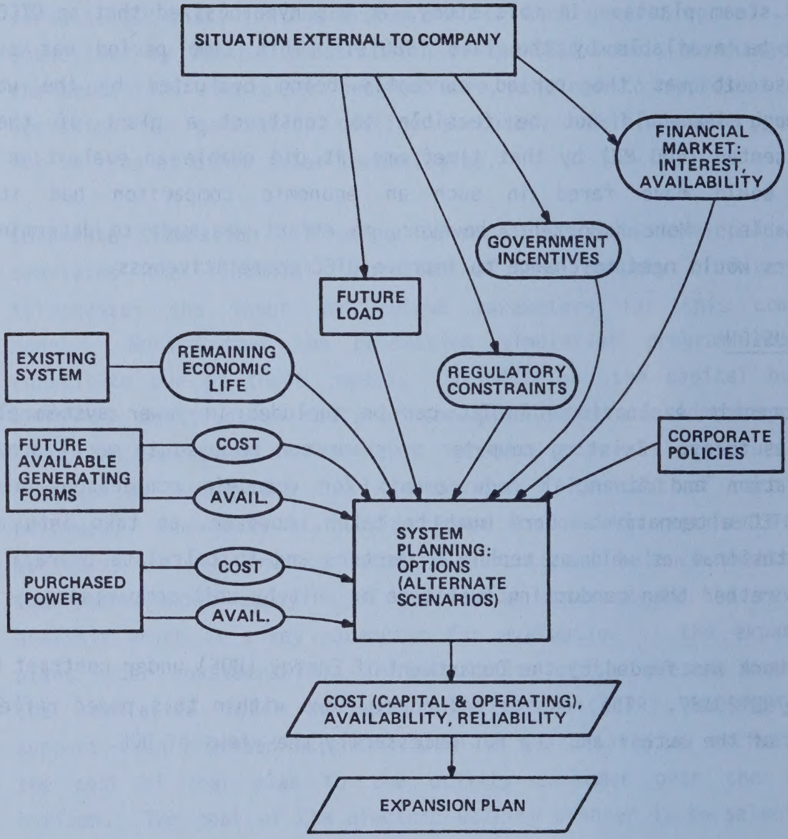
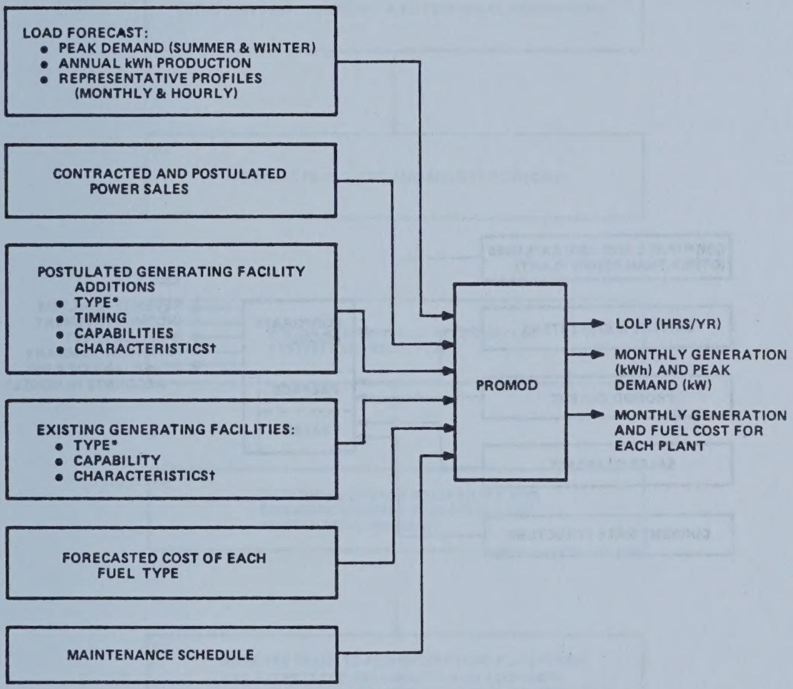

SITUATION EXTERNAL TO COMPANY
FINANCIAL MARKET: INTEREST, AVAILABILITY
GOVERNMENT INCENTIVES
FUTURE LOAD
EXISTING SYSTEM
REMAINING ECONOMIC LIFE
REGULATORY CONSTRAINTS
FUTURE AVAILABLE GENERATING FORMS
COST
AVAIL.
CORPORATE POLICIES
SYSTEM PLANNING: OPTIONS (ALTERNATE SCENARIOS)
PURCHASED POWER
COST
AVAIL.
COST (CAPITAL & OPERATING), AVAILABILITY, RELIABILITY
EXPANSION PLAN

FIGURE 1 PLANNING CONSIDERATIONS


LOAD FORECAST:
• PEAK DEMAND (SUMMER & WINTER)
• ANNUAL kWh PRODUCTION
• REPRESENTATIVE PROFILES (MONTHLY & HOURLY)
CONTRACTED AND POSTULATED POWER SALES
POSTULATED GENERATING FACILITY ADDITIONS
• TYPE*
• TIMING
• CAPABILITIES
• CHARACTERISTICS†
EXISTING GENERATING FACILITIES:
• TYPE*
• CAPABILITY
• CHARACTERISTICS†
FORECASTED COST OF EACH FUEL TYPE
MAINTENANCE SCHEDULE
PROMOD
LOLP (HRS/YR)
MONTHLY GENERATION (kWh) AND PEAK DEMAND (kW)
MONTHLY GENERATION AND FUEL COST FOR EACH PLANT

*PURCHASED CAPACITY OR COMPANY-OWNED FACILITY.

†SEASONAL CAPABILITY; MAINTENANCE OUTAGE REQUIREMENTS; ACCEPTABLE FUELS; COMMERCIAL OPERATION, "MATURITY" AND RETIREMENT DATES; HEAT RATE AND AVAILABILITY (UP TO FIVE CAPACITY LEVELS).

FIGURE 2 PROMOD COMPUTER PROGRAM AS USED IN SYSTEM PLANNING STUDIES

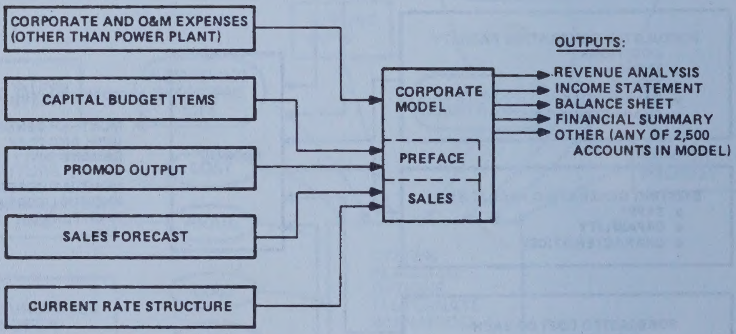

CORPORATE AND O&M EXPENSES (OTHER THAN POWER PLANT)
CAPITAL BUDGET ITEMS
PROMOD OUTPUT
SALES FORECAST
CURRENT RATE STRUCTURE
CORPORATE MODEL
PREFACE
SALES
OUTPUTS:
REVENUE ANALYSIS
INCOME STATEMENT
BALANCE SHEET
FINANCIAL SUMMARY
OTHER (ANY OF 2,500 ACCOUNTS IN MODEL)

FIGURE 3 CORPORATE MODEL COMPUTER PROGRAM AS USED IN SYSTEM PLANNING STUDIES

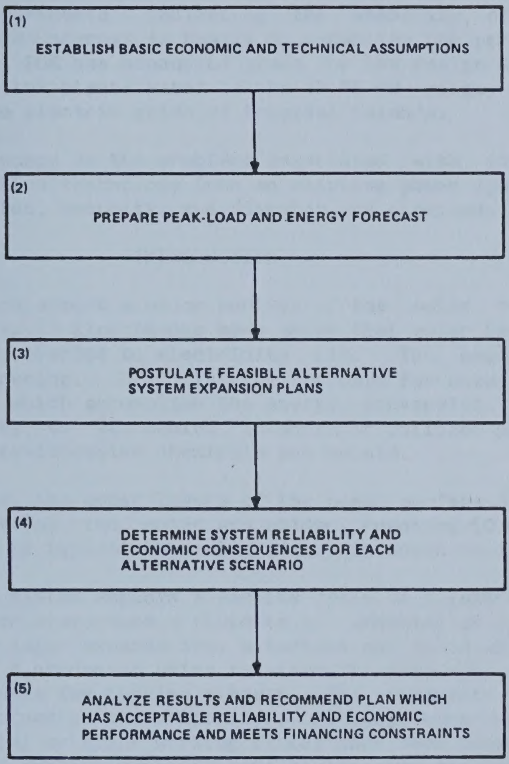

(1)
ESTABLISH BASIC ECONOMIC AND TECHNICAL ASSUMPTIONS
(2)
PREPARE PEAK-LOAD AND ENERGY FORECAST
(3)
POSTULATE FEASIBLE ALTERNATIVE SYSTEM EXPANSION PLANS
(4)
DETERMINE SYSTEM RELIABILITY AND ECONOMIC CONSEQUENCES FOR EACH ALTERNATIVE SCENARIO
(5)
ANALYZE RESULTS AND RECOMMEND PLAN WHICH HAS ACCEPTABLE RELIABILITY AND ECONOMIC PERFORMANCE AND MEETS FINANCING CONSTRAINTS

FIGURE 4 GENERATION FACILITY EXPANSION PLANNING PROCEDURE

DYNAMIC INTERACTION BETWEEN AN OTEC POWER PLANT AND A POWER GRID

ABRAHIM LAVI
ERDI, Inc.
Pittsburgh, Pennsylvania

ABSTRACT

Over the past five years, the prospects for exploiting the ocean thermal gradient has advanced from an R&D curiosity to a serious national program. The U.S. Department of Energy (DOE) has completed a number of studies and experiments indicating the viability of the concept. Presently, tests are underway in Hawaii to establish the performance of key subsystems. Also, DOE has announced plans for the design and construction of demonstration pilot plants rated in the 10-40 MW range. These plants will be tied to the electric grids of tropical islands.

This paper focuses on the problems associated with introducing this new power generation technology into an existing power system. Issues of stability, regulation, security, and dispatch are examined.

INTRODUCTION

Tropical waters absorb a major portion of the solar radiation which the earth receives. Experiments have shown that solar heat collected by the oceans can be converted to electricity [1]. The magnitude of this resource is staggering. OTEC is the acronym for ocean thermal energy conversion systems which accomplish the energy conversion. OTEC systems produce electricity to be cabled to shore or utilized on board for the production of energy-intensive chemicals and metals.

In the tropics, the upper layers of the ocean surface are warm (~28C). The deeper layers of the ocean are colder, reaching 5C or less at 1 Km below the surface. A typical thermal gradient is shown in Fig. 1.

An OTEC power system employs a Rankine cycle as illustrated in Fig. 2. Warm surface water evaporates a fluid (e.g. ammonia) in a heat exchanger (evaporator). The vapor expands into a turbine and parts with some of its thermal energy. A condenser which receives the deep cold water through a long conduit condenses the turbine exhaust. The condensate is pressurized before it is returned to the evaporator. Although other cycles employing seawater (open cycle) or other working fluids have been proposed, for the present, the closed Rankine cycle appears to have the greatest promise. In 1979, a closed-cycle system was tested off Hawaii. The turbine produced 50 KW. Currently, DOE is testing a pair of 1 MW heat exchangers. Other tests of major subsystems are either underway or planned. (See 7th OTEC Proceedings, Washington D.C., June 1980).

Progress in OTEC development has generated a strong momentum for a major system demonstration. DOE has announced plans to procure one or more systems rated at $\geq$ 10 MWe (net) to be connected to the electric grid of an island such as Hawaii, Puerto Rico, or Guam. However, connecting an OTEC power plant to an island grid is not simply a matter of stringing transmission lines from an OTEC platform floating at sea to the electric grid on land. OTEC possesses unique features which may complicate the task of integrating the plant into the grid. Looking beyond the small demonstration plants (10 MWe), we aim to identify and resolve potential problems arising from OTEC plant-grid interactions. In this paper, we describe the characteristics of an OTEC power system that influence the operation of an electric grid. We examine issues of stability, voltage regulation, and economic dispatch of a power system which includes a major OTEC installation. Our work is at an early stage and therefore, we are not yet in a position to offer solutions to these problems. Our aim is to define the problems and to explain our approach.

PROBLEM DESCRIPTION

Most OTEC studies and efforts to integrate OTEC to an existing power network have focused on the cost of OTEC power relative to the cost of power from conventional sources. Only cursory attention has been paid to the compatibility between an existing power network and an OTEC plant supplying a substantial portion of the load. DOE/EES has recently contracted with Florida Power Corp. (FPC) and with the Puerto Rico Power Authority (PRPA) to conduct economic studies and generation planning involving OTEC. However, these studies are basically steady-state analyses. They are not concerned with the short term system dynamics caused by load switching, faults and generation switching. It is not difficult to see why system dynamics and network stability have received little or no attention. First, the economics of the OTEC technology had to be established. Second, to date, no one has developed an authoritative analytical model for the simulation and control of an OTEC power cycle. The first issue will be answered by the FPC and PRPA studies. The second issue has been addressed by Hetyei [2] and Mokhtar-Zadeh [3]. Hetyei has modeled and simulated a total OTEC power system. Although his work has focused on vertical fluted-tube exchangers, it is sufficiently detailed to permit system characterization for transient modeling and control. Recently, Mokhtar-Zadeh developed models for all major candidate OTEC exchangers. The next step is to examine network dynamics involving OTEC installations.

It should be realized that the installation of conventional generating units is always preceded by detailed engineering studies. Such studies, done routinely by electric utilities and their consultants, insure that a proposed plant, its associated transmission lines and its mode of operation will not degrade the performance of the existing power system. Installation of an OTEC plant may introduce new and unforeseen problems which, if not properly identified and resolved, may degrade an otherwise reliable power system.

The dynamic characteristics of an OTEC plant are distinctly different from those of a fossil or nuclear plant. The OTEC turbogenerator inertial constant is likely to be higher on a per kw basis because OTEC turbines operate with a much smaller pressure difference than do conventional steam turbines. The warm and cold water pumps consume about 30% of the OTEC generator load. Connection between OTEC and the network is likely to be at a single point. Thus, a fault or an interruption in the cable could constitute a major shock to the power system. Also, the transmission, whether a.c. or d.c., poses new problems. For instance, with a.c., the relatively high impedance level of the cable may affect network dynamics. Also, if start-up power is supplied by the network to OTEC, heavy a.c. motor loads will result. This added motor load may modify the network characteristics and may present unique problems to the dispatch system. When d.c. transmission is used for non-island applications (with one way power flow), conversion equipment could complicate network response because of the need for special protection. Individually, these problems may not prove significant if OTEC output is small compared to network capacity. Taken together, these problems may become significant even if OTEC delivers a small portion of the load. On the other hand, some of these problems may prove significant if the OTEC output is large compared to network capacity. In addition, if the transmission lines are deliberately over-sized (low impedance) to allow for future expansion of the OTEC installation, network problems may arise even if OTEC output is small.

TECHNICAL APPROACH

Power system engineers have developed computer programs designed to allow short-term simulation of transient phenomena. These programs have displaced a.c. calculating boards because digital simulation offers a wider selection of system parameters and a greater flexibility in network configuration than do the old analog methods. Generally, utilities have a clear idea of their network characteristics -- transmission impedances, generator inertias and reactances, load locations and magnitudes, and inter-generator transmission losses. The data are painstakingly assembled over the years and are continuously updated. Extensive simulation and fault analyses are carried out before a major change in network topology, generation installation, or load addition is contemplated. At the present stage of OTEC development, this degree of detail is unnecessary. In fact, the detail may even hinder our overall understanding of the potential effects of OTEC on the power system and vice versa. What is needed is a model which retains the key features of a network yet simple enough to highlights a macro view of system behavior and to allow isolation of troublesome features. To insure that the results of a simplified model are credible, two precautions must be taken:

1. Develop a simple model (based on detailed data) and exercise the model using different real system disturbances;

2. Test each significant case identified in the simple model simulation by exercising the detailed digital program.

Tentatively, a simplified power network model may consist of two generating units of unequal incremental generation cost and three loads separated by transmission lines. This model constitutes the basic network prior to OTEC installations. An OTEC power plant is then connected to the network through appropriate transmission lines. Recognizing the large number of parameters in the real power system, we intend to use the simple model for the exploration of wider parameter ranges and more OTEC-related fault disturbances at lower computation cost than would be the case if the detailed and specific network model were to be used throughout.

We seek a generic system for analyzing the behavior of a network containing OTEC and eventually, other innovative power generation systems. We are not aiming at investigating generation planning for a particular utility. Our hope is that the results will be universally valid. We shall demonstrate their validity in the case of Puerto Rico and the island of Hawaii.

OTEC-RELATED PROBLEMS

In this section, we elucidate OTEC-related problems.

Impedance Level

The network impedance level will change after the introduction of an OTEC power plant. One source of change is conventional. Whenever a new generating plant is connected to a network through new transmission lines, the impedance level and likewise the natural frequencies of the network change. The location of the tie from OTEC to the network influences the network natural frequency. Another source of impedance change arises from the presence of large synchronous motor loads of the OTEC water pumps. Although speed synchronization of the pump drives with the network is not essential, loss of synchronization is objectionable because of the introduction of oscillations that may trip safety relays.

Start-Up Power

To start an OTEC plant, power is needed to drive the water pumps. In the steady state, pumping power constitutes about 30% of generator output. Connection to the network through an a.c. line will permit OTEC start-up using network power instead of on-board diesel power. But the resultant motor load can be substantial. This is a unique feature of OTEC plants. Hence, the dispatcher must take this condition into consideration. Scheduling of start-up and prior alert may be necessary.

An OTEC installation consists of a number of power modules, or mini power plants, all sharing in water supply and load. OTEC's modular design results in a reduction of the start-up power required because of the feasibility of bootstrapping. However, large systems may consist of proportionately large modules. Power system designers have recommended module power ratings as high as 50 MWe per module, resulting in pumping power of 15 MWe. However, each motor load is not likely to exceed 3.75 MWe with two motors for each evaporator and condenser module. For a 10 MWe power module, a single moter per exchanger will be required and the rating will be 1.5 MWe. Should pump drive be through a variable speed d.c. motor or through a variable frequency a.c. motor, the load characteristics will be different and the impact on the network less significant.

Single Point Tie

With conventional generating units, a line fault is not likely to degrade system synchronization because an alternate path may exist between the generator and the rest of the network. With OTEC, only a single path is practical. A fault in the main transmission line means the severance of the only link between OTEC and the network. Thus, it may be economical to install an underwater cable with spare conductors in order to enhance overall system reliability.

Transmission Cable Impedance

Even though early OTEC plants connected to a network may have a low power rating, it is conceivable that for economic reasons, a high capacity transmission line may be installed. After the early experimental OTEC is proven, the installation may be upgraded by adding more power modules to the OTEC platform. This approach will lead to a low impedance transmission line, oversized for the early OTEC. The system study must take this possibility into consideration.

OTEC Plant dynamics

Because OTEC is a low efficiency system and because its turbine operates at a small pressure ratio (approx. 1.3), turbine inertia is relatively high in comparison to that of steam turbines of the same rating. From a stability viewpoint, this is a welcome feature if OTEC is the only source of power. The OTEC plant will be the sole determinant of system frequency. On the other hand, if OTEC supplies only a portion of the load, loss of synchronization coupled with OTEC's slow response and heavy motor load could make resynchronization both lengthy and difficult.

For a number of reasons, turbine speed control will be effected through a bypass valve across the turbine. The massiveness of the valve, its time delay, and the higher per unit turbine inertia all combine to lengthen the reaction time of OTEC. The ramification of the slow response feature on the dispatch is obvious. The effect on network stability and reliability is unknown at this time.

Output Variation

OTEC is conceived as a baseload power source. This means that the system dispatcher will attempt to operate OTEC at the maximum possible output. Disregarding seasonal and daily temperature variations of seawater, we recognize other factors which may modulate plant output. At times, seawater temperatures may exhibit spurious fluctuations in temperature. Although the average temperature over periods as short as an hour remain steady, ocean currents may cause appreciable instantaneous changes. Temporary changes in intake water temperature may change the output. Since the bypass valve will be fully closed most of the time to produce maximum output, disturbances in the water intake may be transmitted to the network. Conceivably, the thermal inertia of the exchangers and the high inertia of the turbogenerator can partially filter out these disturbances or at least reduce their severity.

Another source of disturbance unique to OTEC stems also from the ocean environment. Platform motion may affect heat transfer in the evaporator. The motion will be reflected in generator output and transmitted to the network. Ingestion of marine organisms by the water pumps and periodic cleaning of debris from inlet screens are likely to modulate plant output. Of course, other (non-OTEC) generating units in the power network are continuously adjusted to meet load demand changes. However, the addition of OTEC-induced variations may tax the conventional units, thereby resulting in unacceptable network hunting at the frequency of OTEC-induced variations. This property is not desirable in a "baseload power source." In fact, this feature may force system planners to choose a generating mix which de-emphasizes OTEC in order to achieve a given system reliability level.

Economic Dispatch

The preceding analysis highlights major differences between OTEC and conventional power plants. Such differences have a direct impact on dispatch strategy. As illustrated in Fig. 3, the dispatcher senses network conditions -- voltage, power, reactive power -- and then issues commands to the various generating stations in order to minimize generation cost and maintain throughout the system voltage and frequency within allowable ranges. The steady-state dispatch problems of a small conventional system are usually straightforward. This would be the case for OTEC if it were to possess characteristics akin to conventional plants. Unfortunately, this is not the case. Hence, it will be necessary to examine the consequences of standard dispatch procedures on network operation when an OTEC plant is connected to the system. The transient disturbances produced by an OTEC plant may require a totally new dispatch strategy.

SOLUTION METHODOLOGY

Model Building

The first task is to combine two or more generating units into a single unit. Consider two generators connected in parallel as shown in Fig. 4. Each generator has its own synchronous voltage, synchronous reactance, and line impedance. The equivalent generator is derived from Thevenin's theorem. The inertia of the equivalent machine is the sum of the inertias of the individual machines.

Each generator has known input-output characteristics: power-out vs. fuel-in. The incremental generation cost is the slope of the input-output characteristic curve multiplied by the cost of the fuel. If more than one machine is combined to form an equivalent generator, the incremental cost function is obtained by adding the power output from all generating units at corresponding incremental cost. The procedure is illustrated graphically in Fig. 5.

The dispatch process may be summarized as follows. For a network consisting of m-generators and k-load centers, determine the output of each generator so as to minimize power generation cost while maintaining load voltages within a prescribed range. Elimination of the voltage constraint considerably reduces problem complexity. The cost minimization problem then reduces to operating all generators at the same incremental cost point on their respective curves. With the use of the simplified model, it will not be necessary to eliminate the voltage constraint. In turn, the model will allow us to use a more sophisticated optimization technique than will the equal incremental cost method. The reason is twofold. First, the two-generator network is easy to handle rigorously. Problems of matrix inversion, complex algebra, and power computation become straightforward. Second, we shall be developing computer routines in connection with network stability studies. These routines are applicable to the dispatch problem using more sophisticated optimization methods.

Mathematically, the dispatch problem may be stated as follows: select p to minimize

$$J = w'p$$

subject to

$$p'Bp + P = p'U$$

$$V_{min} \leq V \leq V_{max}$$

where p is the power generation vector, w cost of generation (usually a function of p), B loss matrix, P load power, U unit vector, V node voltage vector, and the (') denotes transpose. Through the network equations, V and p may be related. In addition, a procedure for network simplication will be necessary. Having combined generation units and remote loads into clusters, we can proceed to simplify the transmission network by a series of delta-wye and star-mesh transformations as is commonly done in a.c. circuit analysis.

Tools Development

Any power system may be described by m-generating units, n-load centers, and k-voltage loops $[k < (m + n)]$. Given transmission, internal and load impedances, a system impedance matrix Z may be derived so that loop current I can be related to generator voltages E by

$$E = Z I \qquad \text{or} \qquad I = Y E$$

where Y is the inverse of Z.

Node voltages are related to the currents by

$$V = A I$$

where A is a rectangular matrix of impedances

The load (complex) power

$$L = P + j Q$$

is computed from

$$L = [H I] \tilde{V}$$

where $\tilde{V}$ is the complex conjugate of load voltage. The product [H I] converts loop currents into load currents. Thus, H is a non-square dimensionless matrix of complex numbers. Note that L is a matrix but we are interested only in its diagonal elements.

Similarly, we can calculate generator output

$$G = I \tilde{E}$$

where G is also a complex matrix containing real and imaginary power.

Any open circuit, fault, or load change modifies the Y, Z, A, B, and H matrices and requires a new steady-state calculation. We need to develop the computer programs that can compute the necessary matrices and circuit conditions for any change in circuitry. Given the power from each generator, the swing angle for each generator is given by

$$M \frac{d^2\delta}{dt^2} = P_\alpha$$

where M is the inertia constant of a machine and P_α is the accelerating power or the difference between mechanical input and electrical output after each has been corrected for losses. The differential equation for δ is nonlinear. Because P_α depends on δ through the function $[\sin(\delta)]$. Because of the nonlinearity, the system of differential equations must be integrated numerically, using the δ's calculated from a.c. circuit analysis as initial conditions.

We have been developing a number of computer programs to solve many of the problems mentioned. We have access to the digital computer codes of the Advanced System Technology Division of Westinghouse Electric Corporation.

CONCLUSION

An OTEC power system introduces new and interesting network problems when the plant feeds power to an existing grid. The unique dynamic characteristics of an OTEC plant ant its ocean environment are the major source of difficulty.

REFERENCES

1. Lloyd Trimble, 7th OTEC Conference, Wahington DC, June 1980

2. Steven A. Hetyei, Ph.D Thesis, Carnegie-Mellon University, 1979

3. M. Mokhtar-Zadeh, Ph.D Thesis, Carnegie-Mellon University, 1980

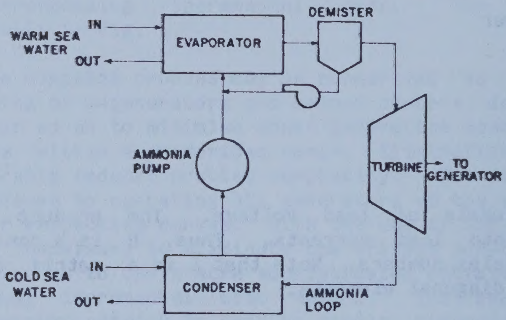

WARM SEA WATER
IN
OUT
EVAPORATOR
DEMISTER
AMMONIA PUMP
TURBINE
TO GENERATOR
COLD SEA WATER
IN
OUT
CONDENSER
AMMONIA LOOP

Fig. 1 A schematic of a closed Rankine cycle for OTEC

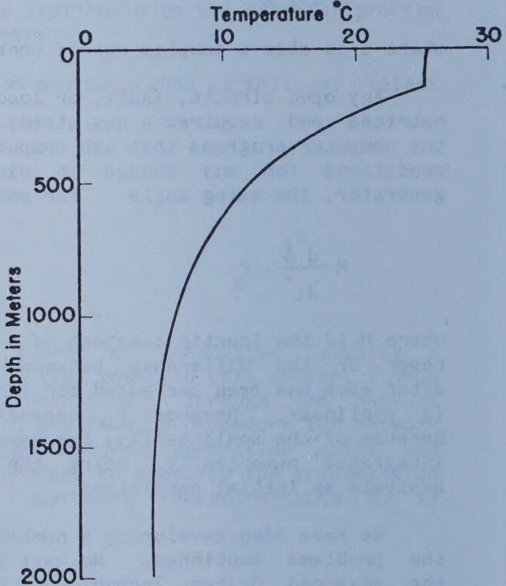


Fig. 2 A typical temperature profile in the tropics

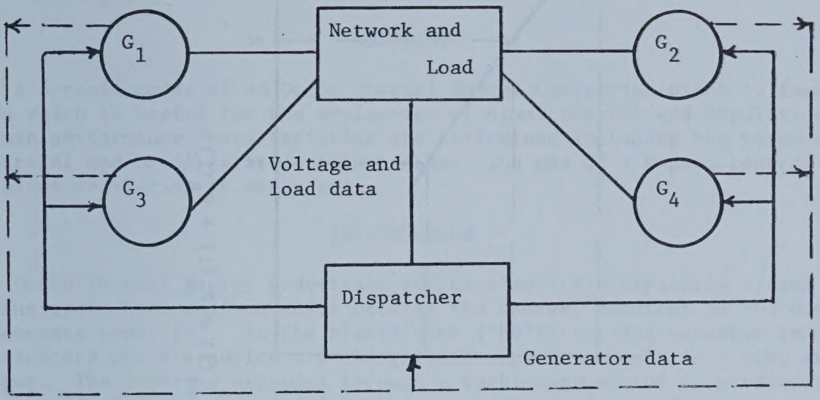
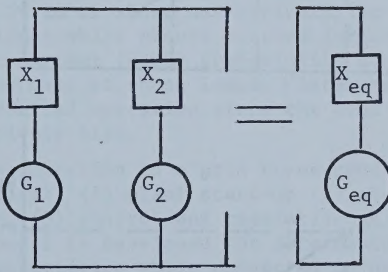
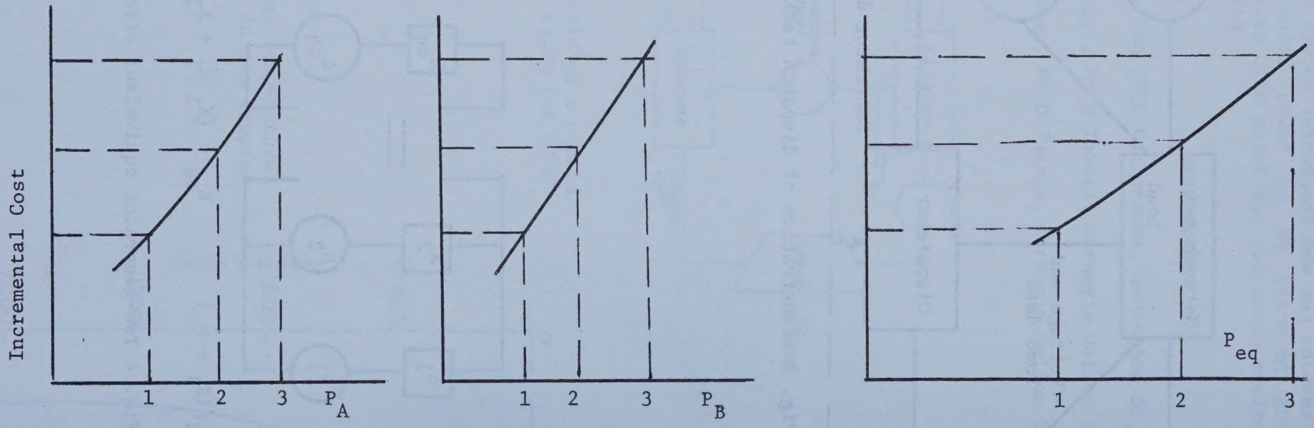

G1
Network and Load
G2
Voltage and load data
G3
G4
Dispatcher
Generator data

Fig. 3 Illustration of dispatch task


X1
X2
Xeq
G1
G2
Geq

$$X_{eq} = X_1 X_2/(X_1 + X_2) \qquad E_{eq} = (X_1 E_1 + X_2 E_2)/(X_1 + X_2)$$

Fig. 4 Two-generator equivalent circuit



$$P_{eq}(i) = P_A(i) + P_B(i)$$

Fig.5 Incremental cost vs. output for equivalent generator

OTEC DYNAMIC SIMULATION MODELS

DAVID N. WORMLEY and GREGORY THOMAS
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

A dynamic model of an Ocean Thermal Energy Conversion plant is formulated which is useful for the evaluation of plant control and regulation. Dynamic performance characteristics are determined including the response to electrical and ocean related disturbances. The use of a bypass control valve for plant regulation is evaluated.

INTRODUCTION

Ocean Thermal Energy Conversion (OTEC) plants are typically closed Rankine cycle heat engines which utilize the thermal gradient in the ocean to generate power [1]. In the plants warm (~80°F) surface seawater is used to evaporate and pressurize a working fluid such as ammonia in a heat exchanger. The vapor is expanded through a turbine-generator to produce electric power and then condensed in a heat exchanger using cold seawater (~40°F) drawn from a depth of approximately 3000 ft. The low pressure liquid condensate is repressurized by a feedpump and returned to the evaporator to complete the cycle. The major OTEC plant components are illustrated in Figure 1.

In design and optimization studies of OTEC plants [2-4] considerable attention has been devoted to evaporator and condenser heat exchanger development since these components operating on small temperature differences (~40°F) along with the turbine-generator are significant factors in plant performance and economics.

One of the areas which must be addressed in OTEC plant development is the plant control and operation when integrated into a utility grid. Plants which are located within 50 km of shore may typically be connected to the utility grid through AC lines while plants located farther from shore require high voltage direct current (HVDC) transmission since AC underwater transmission becomes uneconomic at these longer distances. OTEC plants are proposed primarily for baseload operation since the cost of fuel is low and the capital cost is relatively high.

In consideration of connection to a grid three control and operating conditions may be identified: (1) plant start-up (2) steady-state operating point optimization, (3) control and regulation at an operating point. In this paper a dynamic model is developed for determining OTEC plant control and regulation characteristics while connected to an electrical grid subject to both ocean and electrical disturbances. The model is useful for studies of the dynamics of the plant interaction with the grid and with local plant regulation at an operating point.

The proposed primary control of plant output [2,3] is with a turbine bypass valve which bypasses vapor from the evaporator output around the turbine directly into the condenser. The use of the bypass valve for plant control is considered in detail.

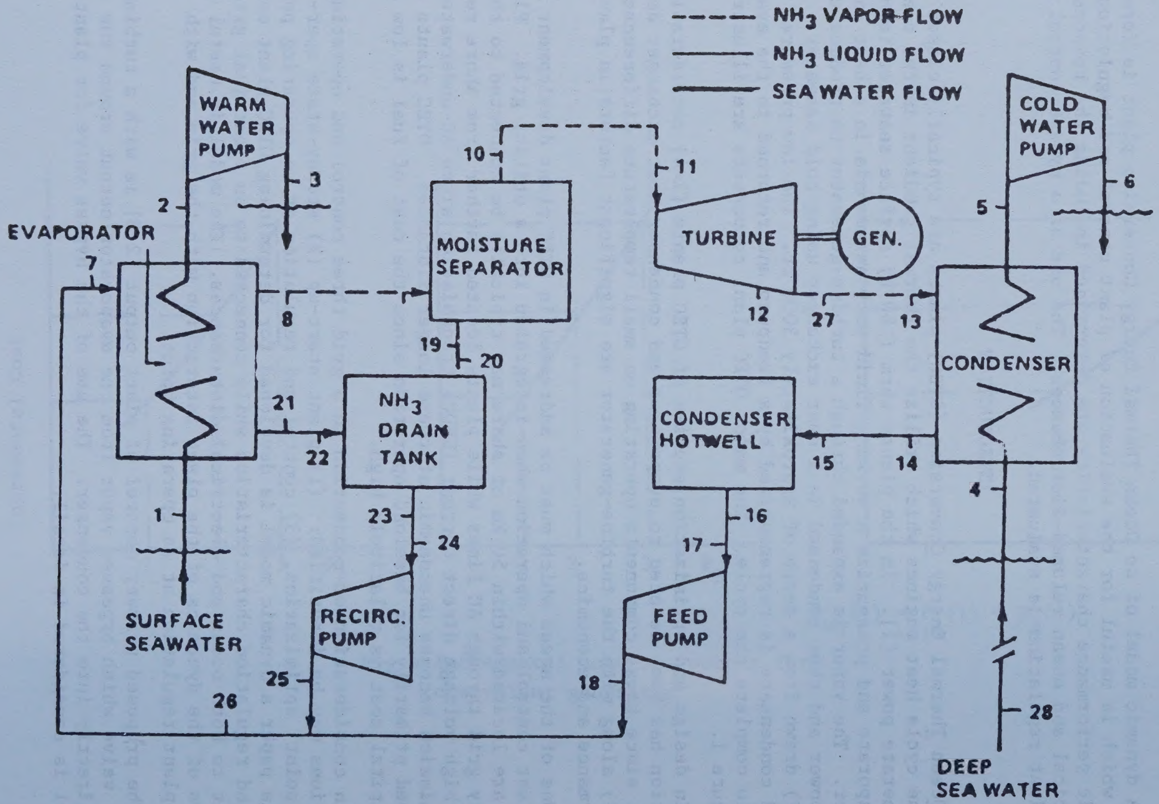

NH3 VAPOR FLOW
NH3 LIQUID FLOW
SEA WATER FLOW
WARM WATER PUMP
COLD WATER PUMP
EVAPORATOR
MOISTURE SEPARATOR
TURBINE
GEN.
CONDENSER
NH3 DRAIN TANK
CONDENSER HOTWELL
RECIRC. PUMP
FEED PUMP
SURFACE SEAWATER
DEEP SEA WATER
1
2
3
4
5
6
7
8
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

FIGURE 1 OTEC PLANT COMPONENTS

MODEL FORMULATION

In this paper a lumped parameter, physically based model of an OTEC plant is developed using mass, energy and momentum balances for each subsystem. The model is of small order focusing upon the dominant system dynamics and has been developed to complement a high order detailed nonlinear model [5] currently under development and other models described in the literature [6,7]. The model parameters and specific simulation results are developed for proposed 50 MW power modules, eight of which would be coupled together to form a 400 MW plant [1].

The model for the OTEC plant is formulated by analyzing the following subsystems - (1) the evaporator, (2) the condenser, (3) the turbine-generator, (4) the cold water system, (5) the warm water system, (6) the vapor transport system, (7) the seawater pump-motor and (8) the generator-electrical load system. The subsystem models are coupled to form the complete model. The detailed model formulation is described in reference [8] and only the primary features are summarized in the paragraphs below.

Evaporator

The evaporator is a horizontal shell and tube heat exchanger which uses warm seawater passing through tubes to evaporate ammonia in the shell. The ammonia vapor leaves the shell through a liquid separator which removes entrained liquid that is returned to the feedpump along with unvaporized ammonia from the evaporator. The model for the evaporator is formulated by assuming that the heat exchanger can be represented by a single equivalent lumped parameter element in which the water and tube metal heat capacities may be combined. This model is based upon an evaluation of the heat exchanger geometric and heat transfer properties which indicates a single lump is sufficient for reproducing the system response to typical transient disturbance characteristic times [9]. The water-tube thermal process may be described as:

$$\frac{dT_2}{dt} = Z_1(T_1 - T_2) - Z_2(T_2 - T_{ve}) \qquad (1)$$

where:
T_1 = evaporator outlet water temperature
T_2 = evaporator inlet water temperature
T_{ve} = evaporator vapor temperature
t = time
Z_1, Z_2 = functions of geometry and fluid properties [8].

In the analysis of the vapor, the heat of vaporization is assumed constant, flashing is neglected, the influence of liquid enthalpy is neglected and the boiling dynamics are considered fast compared to other processes. Under the assumption that the vapor is saturated the vapor density ρ_{ve} may be expressed as:

$$\frac{d\rho_{ve}}{dt} = Z_3(T_2 - T_{ve}) - Z_4 \dot{m}_{vo} \qquad (2)$$

where:
$\dot{m}_{vo}$ = vapor mass flow rate
Z_3, Z_4 = functions of geometry and fluid properties [8]

Condenser Model

The condenser is essentially identical in construction to the evaporator. A lumped parameter model for the condenser may be derived following the basic assumptions described above to yield:

$$\frac{dT_5}{dt} = Z_5(T_4 - T_5) + Z_6(T_{vc} - T_5) \tag{3}$$

$$\frac{d\rho_c}{dt} = Z_7 \dot{m}_{vin} - Z_8(T_{vc} - T_5) \tag{4}$$

where: T_4 = condenser inlet water temperature
T_5 = condenser outlet water temperature
T_{vc} = condenser vapor temperature
ρ_{vc} = vapor density in condenser
$\dot{m}_{vin}$ = inlet mass vapor flow rate
Z_5, Z_6, Z_7, Z_8 = functions of geometry and fluid properties [8]

Turbine-Generator Model

The turbine is a single stage, double axial flow, low pressure ratio unit operating at 1800 rpm and designed to yield a gross output of 68 MW. The equation for the turbine torque balance is:

$$J_{tg} \frac{d\omega_{tg}}{dt} = \tau_{tg} - \tau_L - \tau_f \tag{5}$$

where: ω_{tg}, J_{tg} = turbine-generator angular velocity and inertia
τ_{tg} = turbine-generator input torque
τ_L = turbine-generator load torque
τ_f = turbine-generator friction torque

The torque generated by the ammonia flow is determined by assuming the vapor entering the turbine is saturated and the vapor storage in the turbine is negligible:

$$\tau_{tg} = B_1 \frac{\dot{m}^2_{vtg}}{\rho_{ve}} + B_2 \dot{m}_{vtg} \omega_{tg} \tag{6}$$

where B_1 and B_2 are quanties defined in Reference [8] and where the vapor mass flow rate $\dot{m}_{vtg}$ is given by the Stodola law [10]:

$$\dot{m}_{vtg} = k_{tg} \sqrt{1 - \left(\frac{P_{vo}}{P_{vi}}\right)^2} \tag{7}$$

where: P_{vo} = outlet vapor pressure
P_{vi} = inlet vapor pressure
k_{tg} = turbine flow coefficient

The turbine frictional torque τ_f is assumed to be a function of the prescribed electrical load torque.

$$\tau_f = \alpha \, \tau_e \tag{8}$$

where: α = constant

Cold Water System Model

Cold seawater is drawn from a depth of 3280 feet by two 5800 HP pumps operating at the condenser discharge. The pumps overcome both frictional loss and the fluid head due to the density gradient. The baseload plant has eight separate modules operating from the same cold water pipe. When slug flow is assumed in the cold water pipe and the eight units are assumed to operate in a similar manner the equation for condenser water flow is:

$$N I_p \frac{d F_{wcp}}{dt} = H_1 - H_2 - \Delta H_f \qquad (9)$$

where:
F_{wcp} = water flow rate
N = number of modules operating
I_p = cold water pipe inertance
H_1, H_2 = heads at points 1 and 2
H_f = friction head lost

The friction head H_f is determined from standard friction loss coefficients for turbulent flow.

The energy balance for the cold-water pumps is described by:

$$J_{cp} \omega_{cp} \frac{d\omega_{cp}}{dt} = P_{cs} - \frac{H_{cp} F_{wcp}}{2 \eta_{cp}} \qquad (10)$$

where:
ω_{cp} = cold water pump speed
J_{cp} = cold water pump effective inertia
P_{cs} = power supplied to cold water pump
H_{cp} = cold water pump head
η_{cp} = cold water pump efficiency

and where the head and efficiency are represented as polynominal functions of speed and flow [8].

Warm Water System Model

The warm water system circulates warm surface seawater through the evaporator and discharges it 200 ft below the surface using two 4100 HP pumps. Because the system is relatively compact, fluid inertia may be neglected and the equation for the pump speed may be derived as:

$$J_{wp} \omega_{wp} \frac{d\omega_{wp}}{dt} = P_{ws} - \frac{H_{wp} F_{wwp}}{\eta_{wp}} \qquad (11)$$

where:
ω_{wp} = angular velocity of warm water pump
J_{wp} = warm water pump inertia
P_{ws} = power delivered to warm water pump
H_{wp} = warm water pump head
η_{wp} = warm water pump efficiency
F_{wwp} = warm water flow rate

The pump head and efficiency are polynominal functions of speed and flowrate. Equation (11) coupled with the pump characteristics and the piping friction loss equations may be solved to determine the pump speed and the warm water flow rate.

Ammonia Vapor Transport Model

In the ammonia loop, vapor from the evaporator is transported to the turbine or bypassed through a valve directly to the condenser. When it is assumed the vapor is saturated and fluid inertia and capacitance effects are neglected, the mass flow from the evaporator is:

$$\dot{m}_{vo} = \dot{m}_{vtg} + \dot{m}_{vbp} \tag{12}$$

where the bypass valve flow $\dot{m}_{vbp}$ is:

$$\dot{m}_{vbp} = k_{bp} \sqrt{1 - (\frac{P_{vo}}{P_{vi}})^2} \tag{13}$$

where the quantity k_{bp} is the bypass valve flow coefficient and is a function of the valve position. The flow to the turbine $\dot{m}_{vtg}$ is given by Equation (7).

Electrical Generator Model

In this paper the plant is considered to be connected to shore through a high voltage DC link in which the OTEC AC generated voltage is converted to DC, transmitted to shore and then reconverted to AC at the utility grid. When the response times of the DC link and conversion equipment are considered to be fast compared to the OTEC plant response times the electrical torque is:

$$\tau_e = P_{DC}/\omega_{tg} \tag{14}$$

where: P_{DC} = power transmission

Seawater Pump-Motor Model

Each seawater pump is powered by an AC synchronous motor. The power supplied to each motor is represented as

$$P_e = \frac{E_m E_T}{x_m} \sin \delta_m \tag{15}$$

where: E_m, E_T = pump motor field and terminal voltages

x_m = pump motor synchronous reactance

δ_m = pump motor power angle

The pump motor power angle is given by:

$$\frac{d}{dt} \delta_m = \omega_m - \omega_s \tag{16}$$

where: ω_m, ω_s = pump motor angular speed and set point speed

and the motor torque is:

$$\tau_m = \frac{E_m E_T}{\omega_s x_m} \sin \delta_m \qquad (17)$$

The cold and warm water systems are each driven by one equivalent motor unit.

Model Synthesis

The component equations developed in the sections above may be combined into a set of nonlinear, coupled first order differential equations. These equations are summarized in reference [8] as:

$$\frac{d\underline{x}}{dt} = \underline{f}[\underline{x},\underline{u}] \qquad (18)$$

where $\underline{x}$ and $\underline{u}$ are respectively the state and input vectors and $\underline{f}$ is the set of nonlinear functions.

In this study which is focused on operation during disturbances from a steady-state plant operating point, the equations describing the plant are linearized and placed in state equation form. When the nonlinear equations are expressed in terms of variations from the steady state, the following linearized equations may be derived in terms of the variations $\underline{\Delta x}$ and $\underline{\Delta u}$:

$$\frac{d\underline{\Delta x}}{dt} = \underline{A}\,\underline{\Delta x} + \underline{B}\,\underline{\Delta u} \qquad (19)$$

where the matrices $\underline{A}$ and $\underline{B}$, are constant coefficients determined at the steady-state operating point:

$$\underline{A} = \left.\frac{\partial \underline{f}}{\partial \underline{x}}\right|_{ss} \qquad \underline{B} = \left.\frac{\partial \underline{f}}{\partial u}\right|_{ss} \qquad (20)$$

The resulting state vector and matrices $\underline{A}$ and $\underline{B}$ are summarized in reference [8] along with sets of steady-state values used to evaluate the linearized constants for several operating power levels of the OTEC plant.

SYSTEM DYNAMIC CHARACTERISTICS

The system dynamic characteristics have been studied first by determining the system eigenvalues from the $\underline{A}$ matrix to estimate the range of response times inherent in the system. The time constants were determined for the case in which the electrical load to the turbine-generator was represented as a constant electrical load torque and the cold and warm water pump AC motor time constants were neglected. The system eigenvalues for 100% and 60% are summarized in Table 1 where it is assumed the load is varied from 100% by opening the turbine bypass valve appropriately. The data in the Table show that cold water system flow time constant is the longest-26.6 sec. The turbine-generator time constant is 14.1 sec. The evaporator and condenser both have thermal time constants on the order of

TABLE 1

OTEC PLANT TIME CONSTANTS (seconds)

COMPONENT	100%	60% LOAD
Evaporator: Thermal	5.56	5.05
Evaporator: Vapor Flow	0.64	0.55
Condenser: Thermal	7.20	7.14
Condenser: Vapor Flow	0.77	0.77
Turbine-Generator Generator	14.10	15.90
Warm-Water System	0.23	0.23
Cold-Water Pump	0.14	0.14
Cold-Water Flow	26.6	26.6

7 secs. The data at the two load levels show that the system time constants do not vary significantly with load level.

Examination of the system shows that the cold and warm water flow rates are essentially independent of the remainder of the system. Thus, for fixed settings of these flow rates, the OTEC plant dynamics are essentially controlled by the turbine-generator time constant and the evaporator and condenser thermal time constants.

The response of the system to changes in the bypass valve opening, and to changes in wave disturbances are illustrated in Figures 2 and 3. With a 10% opening in the bypass valve, the vapor pressure to the turbine rapidly decreases in 2-3 seconds after which the temperature in the evaporator stabilizes in 20-30 secs. and then the turbine speed stabilizes in 50-60 seconds. The 10% change in valve opening reduces turbine speed by 12.5%.

A wave disturbance corresponding to a sinusoidal change in head of 12 feet amplitude and ten second period causes a variation in warm water flow of about 30% and results in a variation of 1.6% in plant power output when the power delivered to the warm water pump is maintained constant. An additional simulation has shown if the synchronous motor speed is maintained constant, the seawater flow variation is maintained at less than 5%, however, the pump power requirements vary by 100%. This change in pumping power results in a 10% change in net OTEC power output.

OTEC PLANT REGULATION

The primary method available to regulate the OTEC plant is through control of the turbine bypass valve. Control of the cold and warm water flows does not provide sufficiently fast response for regulation since these effects act through the system thermal lags, although control of these flows is desirable for optimization of power at steady-state operation.

The basic control system which has been proposed for OTEC plants is a turbine speed governor which modulates the bypass valve opening. The

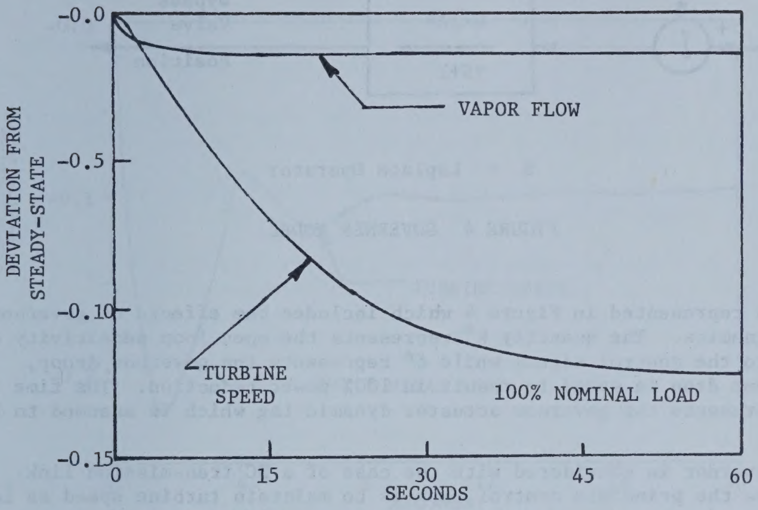


FIGURE 2 TURBINE VAPOR FLOW AND SPEED RESPONSE TO 10% OPENING IN BYPASS VALVE

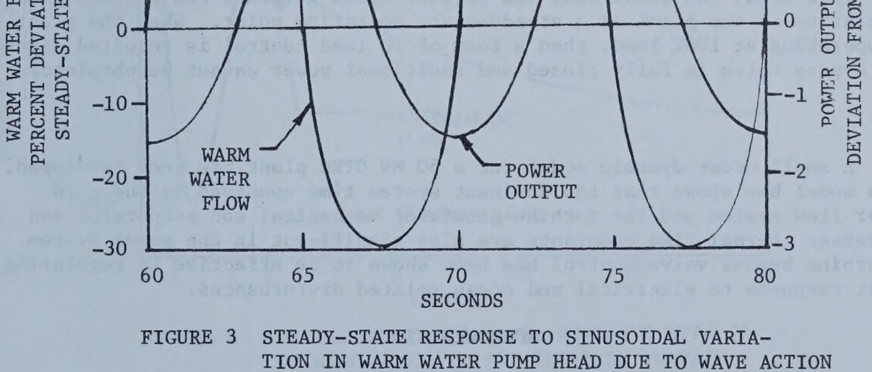
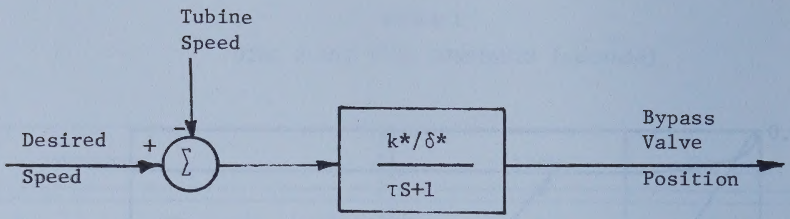


FIGURE 3 STEADY-STATE RESPONSE TO SINUSOIDAL VARIATION IN WARM WATER PUMP HEAD DUE TO WAVE ACTION



S = Laplace Operator

FIGURE 4 GOVERNER MODEL

governor is represented in Figure 4 which includes the effects of governor actuator dynamics. The quantity k^* represents the open loop sensitivity of the valve to the control signal while δ^* represents the governor droop, i.e., percent drop in speed to result in 100% power reduction. The time constant τ represents the governor actuator dynamic lag which is assumed to be 0.2 secs.

The governor is considered with the case of a DC transmission link. In this case the principle control task is to maintain turbine speed as load and disturbances change. A baseline condition of 80% power level is considered with a 5% increase in load demand. Responses were computed for droop settings varying from 1% to 5%, with stability and steady-state error increasing as droop increases. A value of 3% was found to be a good compromise between stability and steady-state error. The response for this condition is shown in Figure 5. The speed droops 0.2% and results in a steady-state error of 0.1% with the dynamic adjustment occurring in 8 secs.

The response of the system to a decrease by 1°F in warm seawater temperature is shown in Figure 6. This figure shows that as the seawater temperature in the evaporator decreases, the vapor flow rate decreases and the power decreases until the speed controller closed the bypass valve and brings the power back to the desired level. This corrective action occurs over a ten-second period and results in less than 0.1% steady-state error.

This study has shown that the turbine speed governor can provide good regulation of the plant at a steady-state operating point. When the plant is operating at 100% load, then a form of DC load control is required since the bypass valve is fully closed and additional power cannot be obtained.

SUMMARY

A small order dynamic model for a 50 MW OTEC plant has been developed. This model has shown that the dominant system time constant is the cold water flow system and the turbine-generator mechanical and evaporator and condenser thermal time constants are also significant in the power system. A turbine bypass valve control has been shown to be effective in regulating plant response to electrical and ocean related disturbances.

ACKNOWLEDGEMENT

The research described in this paper has been supported by the U.S. Department of Energy, Division of Electrical Energy Systems.

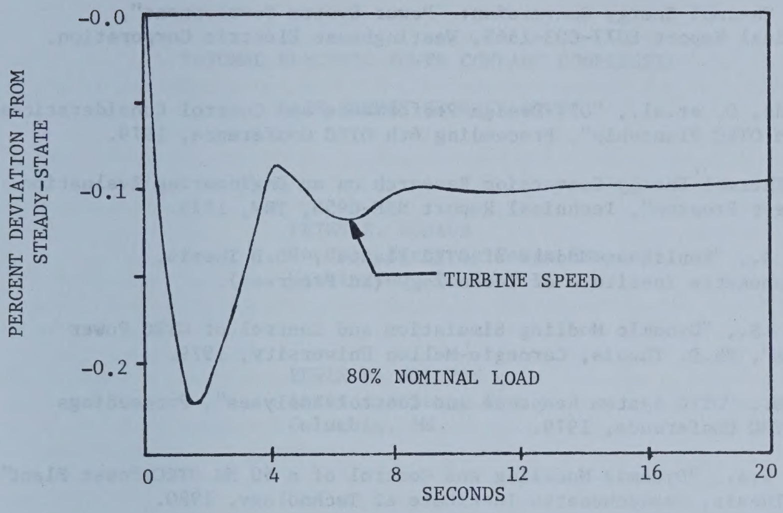


FIGURE 5 TURBINE SPEED RESPONSE TO 5% INCREASE IN LOAD DEMAND WITH 3% GOVERNOR DROOP

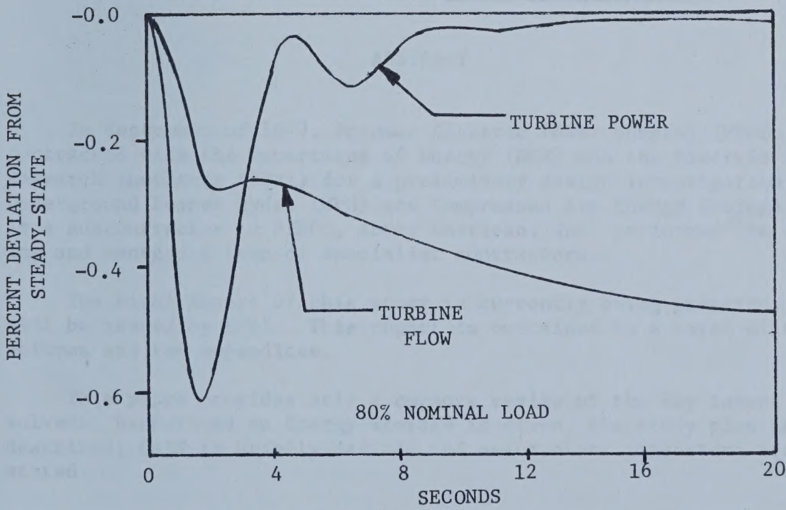


FIGURE 6 TURBINE SPEED AND POWER RESPONSE WITH 3% GOVERNOR DROOP TO 1°F DECREASE IN WARM WATER TEMPERATURE

REFERENCES

1. Carmichael, A.D., "Ocean Thermal Energy Conversion: A State of the Art Study", EPRI Technical Report ER-1113-SR, 1979.

2. "Ocean Thermal Energy Conversion: Power System Development", Technical Report EG77-C03-1569, Westinghouse Electric Corporation, 1978.

3. Richards, D. et.al., "Off-Design Performance and Control Considerations for an OTEC Plantship", Proceeding 6th OTEC Conference, 1979.

4. "Ocean Thermal Energy Conversion Research on an Engineering Evaluation and Test Program", Technical Report NSF-C958, TRW, 1975.

5. Sotelo, J., "Nonlinear Models of OTEC Plants", Ph.D Thesis, Massachusetts Institute of Technology (In Progress).

6. Hetyei, S., "Dynamic Modling Simulation and Control of OTEC Power Plants", Ph.D. Thesis, Carnegie-Mellon University, 1979.

7. Owens, W., "OTEC System Response and Control Analyses", Proceedings 6th OTEC Conference, 1979.

8. Thomas, G.A., "Dynamic Modeling and Control of a 50 MW OTEC Power Plant" S.M. Thesis, Massachusetts Institute of Technology, 1980.

9. Masada, G., "Modeling and Control of Power Plant Boiler-Turbine-Generator Systems", Ph.D. Thesis, Massachusetts Institute of Technology, 1979.

10. Stodola, A., Steam and Gas Turbines, Smith Pubs. New York, 1945.

POTOMAC ELECTRIC POWER COMPANY COMPRESSED AIR ENERGY STORAGE STUDY

PETER E. SCHAUB
Potomac Electric Power Company
Washington, D. C.

EDWIN D. SHIPPEY
Acres American, Inc.
Columbia, Md.

ABSTRACT

In September of 1977, Potomac Electric Power Company (PEPCO) contracted with the Department of Energy (DOE) and the Electric Power Research Institute (EPRI) for a preliminary design investigation of Underground Pumped Hydro (UPH) and Compressed Air Energy Storage (CAES). As a subcontractor to PEPCO, Acres American, Inc. performed the engineering and managed a team of specialist contractors.

The Final Report of this study is currently being prepared, and will be issued by EPRI. This report is contained in a total of thirteen volumes and ten appendices.

This paper provides only a cursory review of the key issues involved. Background on Energy Storage is given, the study plan is described, CAES is briefly defined and major plant characteristics are stated.

BACKGROUND

What is Energy Storage?

Energy Storage is the process by which potential energy is accumulated during a period of abundance and converted to power on demand. A utility's coal pile represents energy storage, but the process of conversion to power is complex and requires an intermediate working fluid. The system has an inertia which is not always easily controlled to match consumer demand. Oil-fired units have less response-time inertia and are frequently used to match the rapid changes and peaks of demand. Their fuel oil tanks are their energy storage devices. Ideally, utilities would prefer a fast-response power plant which does not burn oil.

In the 1960's, this need was met with Pumped-Storage Hydroelectric Stations. These, such as Muddy Run along the Susquehanna River in Pennsylvania, utilized natural terrain and large bodies of water. During peak periods, water was released from an upper reservoir, through turbines to a lower reservoir. Overnight, with consumer power demand low, the turbines were reversed and driven by motor to pump back into the upper reservoir. The flow restrictions of run-of-river hydroelectric plants were circumvented by the recirculation of a fixed volume of water. Planning studies indicated overall economics to lie with the pumped storage, against such alternates as diesel-generators, gas turbines and steam peaking units (1).

PEPCO Involvement

In the late 1960's and early 1970's, PEPCO was planning to grow in system capacity at a rate of about 10% per year. Substantial baseload additions needed balancing by peaking installations. A typical PEPCO summer day load curve is used in Figure 1 to demonstrate the need for peaking power and the value of Energy Storage. In midafternoon the load is twice that at predawn. Baseload plant cannot accomodate both conditions and the rate of change between them. Rate of change and peak demands are met at PEPCO by gas turbines fired with No. 2 fuel oil. These could be replaced by Energy Storage facilities charged by base load capacity off-peak. This prompted PEPCO to undertake investigations of the feasibility of pumped storage.

Pumped Storage requires specific terrain and substantial alteration of land and water resources. Remote sites necessitate rather extended transmission line rights-of-way. Availability of suitable sites, growing environmental awareness and the need for long transmission lines discouraged PEPCO from this option.

About this same time, interest developed in a new proposal: Pumped Storage using a lower reservoir excavated from rock at a significant depth underground (UPH). By utilizing high heads, the water flow and reservoir volumes could be dramatically reduced. In fact, the upper reservoir might be a totally man-made, self-enclosed structure. PEPCO commissioned studies which identified suitable host rock bodies and surface sites for UPH.

Compressed Air Energy Storage was also proposed for these hard-rock sites and was found feasible. CAES is based on the gas turbine cycle with the compressor discharge stored for later use in driving the turbine. Surface and subsurface requirements similar to those for UPH were identified.

The events of 1973-74 brought load growth to a standstill at PEPCO. Much of the base load generation in planning was deferred. The need for peaking power and Energy Storage decreased.

THE "CAUPH" STUDY

The U. S. Energy Research & Development Administration (ERDA, forebear of DOE) and EPRI continued support of UPH and CAES within their research and development programs. PEPCO, among many other utilities, manufacturers and engineering firms, worked with ERDA and EPRI to develop a program which would generically, but definitively assess UPH and CAES for future implementation.

In 1976, a joint ERDA/EPRI program was established and proposals for site-specific investigations were sought from utilities. Underground storage of air (CAES) was proposed for three geologies: salt, with caverns leached out of a salt dome by a controlled water pumping process; aquifers, with air to be injected into the porous media to displace and be contained by the water; and hard rock, with caverns excavated by conventional techniques. UPH was included in the program, but limited by its nature to hard rock sites.

PEPCO sought and received a contract to study UPH and CAES in rock, as a natural continuation of previous in-house studies. Designated "CAUPH" for Compressed Air/Underground Pumped Hydro, the study began in 1977, with Acres American, Inc. serving as engineers to PEPCO. This work has now been brought to successful conclusion. EPRI is in the process of publishing the thirteen volumes and ten appendices which definitively document preliminary designs of CAES in rock, and UPH. These books are included as references no. 2-24.

Middle South Services has recently completed their parallel effort on CAES in salt. EPRI will publish their reports. Public Service Indiana is currently investigating CAES in aquifers under a third DOE contract.

CAUPH Objective and Plan

The stated objective of the study was: "to provide firm estimates of cost, schedule and risks upon which a decision to construct a CAES or UPH facility can be based." The plan to address this objective included five "Tasks", generally sequential:

Task 1 - "Design Criteria" was accomplished in two separate phases. The first was completed very early in the work and was, simply, establishing the overall guidelines for the engineering work and developing bounding assumptions for CAES and UPH facilities. The work consisted of a literature search and survey of precedent machinery and installations; then documentation, in a report, of basic assumptions, codes and governing design methods and formulas for systems and components, classified by engineering discipline. PEPCO, at this stage, performed full expansion planning analyses to establish plant size, storage capacity, installation timing, relative economics and sensitivities to cost and performance characteristics. Just recently as one of the last activities of the study, PEPCO iterated on these studies. The final analyses are reported in (5) and (25). Acres is reviewing and updating the Design Criteria Reports (3, 4) to reflect the changes in our state of knowledge following the preliminary design activity.

Task 2 - "Site Selection and Investigation" was included in the work to force site specificity into the planning and engineering functions. The value of the work is largely attributable to the real-world situations imposed, because these resulted in preliminary designs which address engineering obstacles. The first work step consisted of preparing a site selection methodology and evaluating a group of potential sites (6). Since the key to UPH and CAES is sound geology, a program was next developed to explore the rock by a series of shallow boreholes and one deep borehole. A site suitable for drilling was chosen and both drilling programs carried out. The information gleaned (7, 8) was used in preparing all designs. The drilling site, found suitable for development, was used also for specificity of design.

Task 3 - "Design Approaches" comprised the bulk of the work. Using results from Tasks 1 & 2, the various engineering alternatives for system and component design were weighed and judged. Specialist subcontractors and machinery manufacturers were awarded work in support of Acres' overall effort to establish a coherent configurations for CAES and UPH. Near the end of the task, draft construction schedules and cost estimates were prepared. This work will be available in two "handbooks" for future design: Final Report Vol. VIII (UPH) and Vol. IX (CAES), with their ten appendices (9-20).

Task 4 - "Environmental Studies" was performed under a special subcontract. This work was based on Tasks 1, 2 and 3 and allowed to proceed rather independently of the main design thrust. The result is an analysis of potential impacts and their relative magnitude (21). Further design efforts must address the concerns raised. Public safety issues were also addressed in this task.

Task 5 - "Plant Design" was primarily documentation of coherent designs for UPH and CAES based on Task 3 results. Final design decisions were reached and cost economies achieved. Consideration was given to construction procedures, site development, reliability and cost-risk. Overall control logic was addressed. Final cost estimates and construction schedules were prepared on the bases of final designs. Cash flow information was developed. Volumes XI and XII of the Final Report include component/system descriptions and an entire series of preliminary design drawings (22, 23). Originally conceived as part of the Task 5 report, but eventually requiring a volume of its own, is the Preliminary Licensing Documentation (24). Acres reviewed federal, State of Maryland, and local regulations and developed a plan to approach the licensing of these new technologies in today's regulatory arena.

Key Issues Relating to CAES

During preparation of the proposal for study to ERDA, certain key issues were identified as requiring investigation or development. In mechanical engineering, establishing the storage pressure was of primary importance. This decision was seen to require an optimization of the thermodynamics with available machinery in order to achieve maximum operating economy while not requiring extensive and costly machinery development.

The whole question of designing and excavating the shafts and underground caverns was of importance in civil engineering. Questions of cavern structural integrity and sealing air at high pressure within the system had to be answered.

It was anticipated early that control systems for the CAES plant would be complex in order to satisfy all expected modes of operation. Later in the study, it was found that motor starting was also a design concern.

THE CAES DESIGN

What is CAES?

A schematic of the Compressed Air Scheme is shown in Figure 2. During compression, the expansion turbine is separated from the motor/generator by a clutch. Air leaving the compressor is cooled to about 120°F and routed to storage underground at a pressure around 1000 psi. The heat generated in compression is exhausted to the atmosphere and, therefore, CAES does not store much "energy" in the air at 1000 psi and 120°F.

What is stored is the compressed air in which fuel oil is fired to drive the turbine. In the conventional Brayton (gas turbine) cycle, the compressor is driven by the turbine. About two-thirds of the power generated by the turbine, and two-thirds of its fuel consumption, is used in driving the compressor. With air already in storage, the turbine is declutched from the compressor in CAES, and 100% of its power drives the generator. Stated another way, three times the power can be generated with CAES over gas turbines using the same amount of fuel oil.

Hydraulic compensation of the stored air is provided by a water filled shaft supplied by a small, enclosed surface reservoir. By maintaining constant storage pressure, the storage volume may be kept much smaller than in a variable pressure, or constant volume design. Acres has determined that the savings in excavation cost for the caverns is greater than the cost of the reservoir and water shaft.

Mechanical Design Issues

In 1978, the world's first CAES plant (in salt) went into operation in Huntorf, Fed. Rep. of Germany with rotating machinery supplied by Brown Boveri and Sulzer (26). This experience was rated high and Brown Boveri and Sulzer were engaged to develop a machinery train for the PEPCO application.

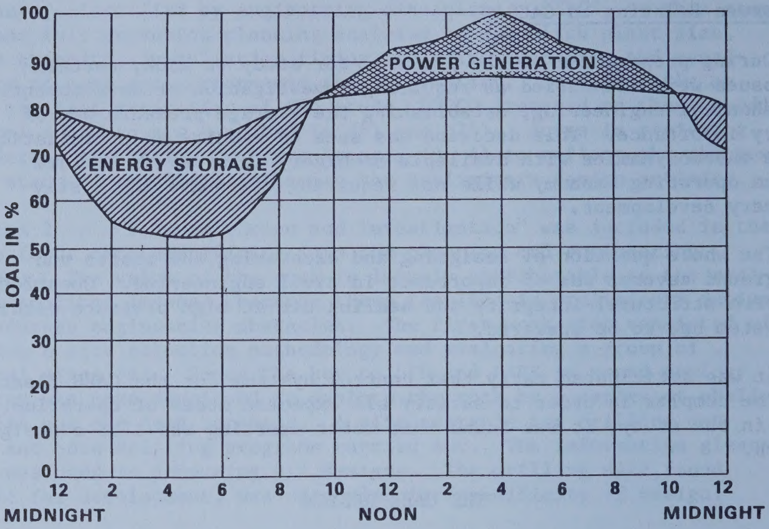
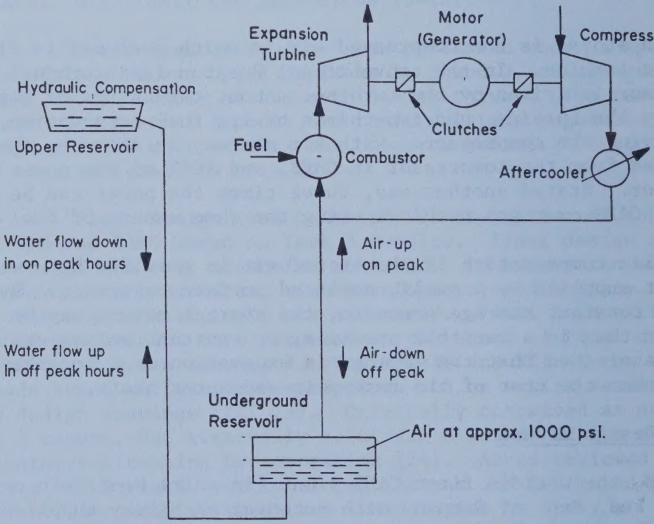


FIGURE 1 - TYPICAL SUMMER DAY LOAD PROFILE:
The demand at 4 a.m. is half that at 4 p.m. By running base-load plant at full load overnight, the storage station can be charged. The stored energy is used to supplement the base-load plant on-peak the next day.



Expansion Turbine
Motor (Generator)
Compressor
Hydraulic Compensation
Clutches
Upper Reservoir
Fuel
Combustor
Aftercooler
Water flow down in on peak hours
Air-up on peak
Water flow up in off peak hours
Air-down off peak
Underground Reservoir
Air at approx. 1000 psi.

FIGURE 2 - COMPRESSED AIR ENERGY STORAGE SCHEMATIC

Thermodynamics would dictate air storage at as high a pressure as possible, since air volume and cavern cost decrease with pressure. Sulzer compressors, however, reach a threshold at about 1000 psi where a barrel type must be used in lieu of a split case design. The associated step increase in cost overcomes the pressure/cost advantage. Hence, 1000 psi storage was adopted.

An outgrowth of the thermodynamic analysis was the need for exhaust gas recuperating during generation. Such a high pressure, high temperature, high flow recuperator has never been built. Nonetheless, its effect on fuel consumption requires it if good operating economics are to be achieved. DOE/EPRI are continuing research into this equipment, beyond the CAUPH study.

Figure 3 is an elevation view of the final machinery train.

Civil Design Issues

Considerable strength testing of rock from the drilling program was performed. This data and the results of in-situ rock strength testing (7, 8) were used in a finite element analysis of cavern stability (14). In parallel a study of excavation techniques and costs was undertaken. The combined design optimization led to the 85ft. high 66ft. wide cavern cross-section shown in Figure 4. Also shown in this figure is a plan view of underground caverns.

Air is sealed within the air shaft by a steel liner (16). Air is prevented from escaping up the water shaft by a U-tube in the bottom of the shaft. A question has been raised whether air dissolved in the water could escape up the water shaft. This is treated in detail in Final Report Vol. IX, Appendix B (17). Briefly, geometry of the shaft/cavern and shaft/reservoir intersections, the U-bend and restrictions on operation are believed to be preventative measures. DOE/EPRI are continuing work in this area also.

Control System Design Issues

The CAES plant must operate as a fast-response peaking facility. Each unit must be self-starting with either emergency or station service power, the units must be able to operate over the entire load range, and synchronous condenser operation must be possible. During Task 3, it was determined that the basic operational capabilities provided by Brown Boveri and Sulzer as part of their standard control packages can be integrated into a coordinated control system for power generation, air compression and synchronous condenser operation equal to the imposed criteria. Normal generation startup to full load can be accomplished in ten minutes, emergency startup in five and a half minutes, and changeover from full compression to full generation in nineteen minutes under emergency conditions (15).

It is possible to start up a CAES unit in the compression mode by running up the entire machinery train with the turbine. Once the motor is synchronized, the turbine can be tripped and automatically declutched from the motor. This method is mechanically fatiguing to the turbine, however, and significantly reduces its life expectancy. The final plant design, therefore, calls for variable frequency

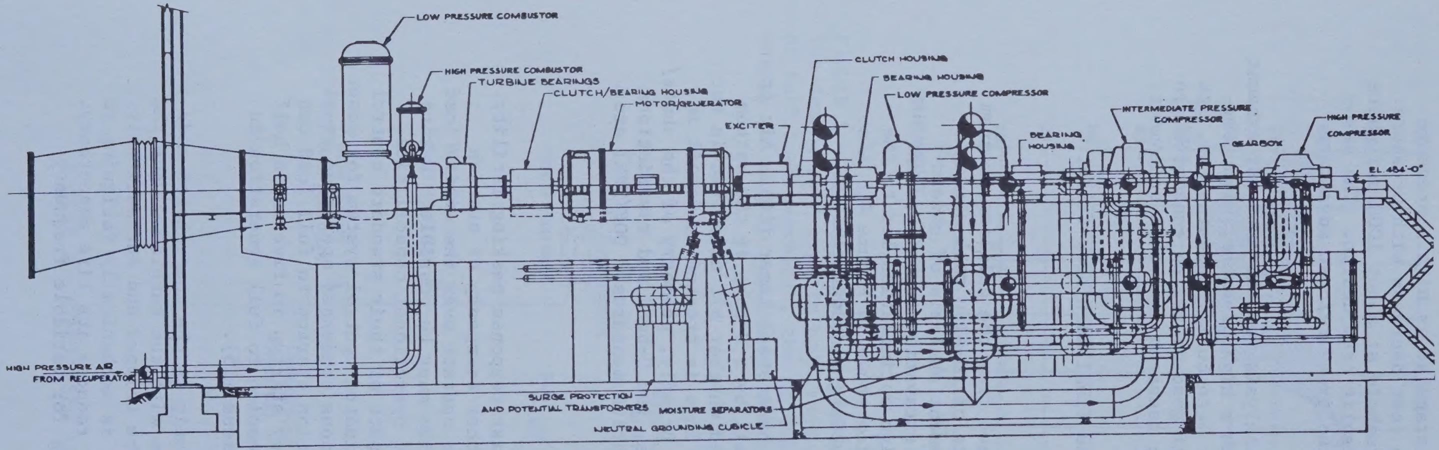
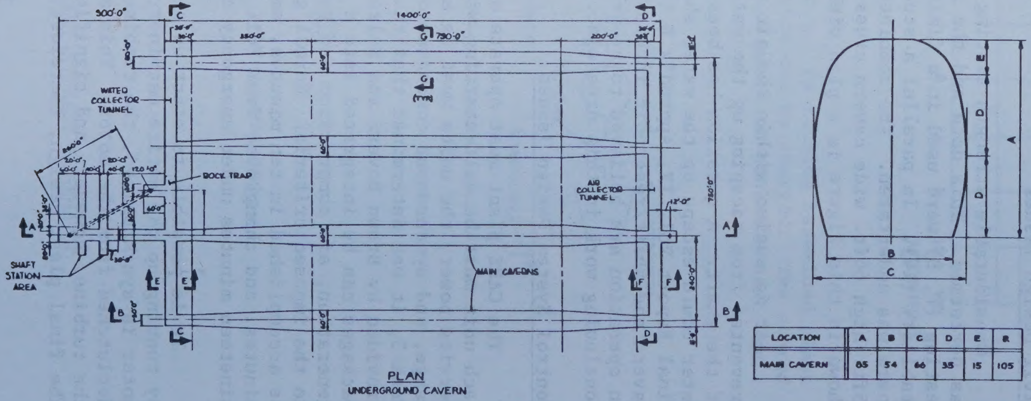

LOW PRESSURE COMBUSTOR
HIGH PRESSURE COMBUSTOR
TURBINE BEARINGS
CLUTCH/BEARING HOUSING
MOTOR/GENERATOR
EXCITER
CLUTCH HOUSING
BEARING HOUSING
LOW PRESSURE COMPRESSOR
BEARING HOUSING
INTERMEDIATE PRESSURE COMPRESSOR
GEARBOX
HIGH PRESSURE COMPRESSOR
EL 484'-0"
HIGH PRESSURE AIR FROM RECUPERATOR
SURGE PROTECTION AND POTENTIAL TRANSFORMERS
MOISTURE SEPARATORS
NEUTRAL GROUNDING CUBICLE

FIGURE 3 (ABOVE) - CAES MACHINERY TRAIN IN ELEVATION: Piping maze on right is between compressors and air coolers. Note HP & LP vertical combustors.


300'-0"
1400'-0"
C
D
35'-0"
30'-0"
350'-0"
G
750'-0"
200'-0"
80'-0"
60'-0"
15'-0"
G
(TYP)
WATED COLLECTOR TUNNEL
260'-0"
20'-0"
40'-0"
120'-0"
ROCK TRAP
60'-0"
AIR COLLECTOR TUNNEL
12'-0"
750'-0"
A
B
E
F
SHAFT STATION AREA
MAIN CAVERNS
240'-0"
PLAN
UNDERGROUND CAVERN
A
B
C
D
E

LOCATION	A	B	C	D	E	R
MAIN CAVERN	85	54	66	35	15	105

FIGURE 4 (LEFT) - CAES STORAGE CAVERNS: Plan view and typical cross section. Main cavern dimensions are at Section G-G.

starting. The PEPCO design includes a static frequency converter requiring a 40 MVA power source (15, 23).

RESULTS AND CONCLUSIONS

Some characteristics of the final PEPCO CAES design are given in Table 1. These are the result of three years' of engineering analysis and preliminary design. So detailed, and so successful was the work, that CAES is a technology which can now be implemented with confidence by U. S. utilities.

REFERENCES

1. Dickinson, W. H., "Muddy Run pumped storage," Electrical World, July 4, 1966, pp. 31-33.

2. "Preliminary Design Study of Underground Pumped Hydro and Compressed Air Energy Storage in Hard Rock," by Potomac Electric Power Comany and Acres American, Inc. for the U. S. Dept. of Energy and Electric Power Research Institute, 1980. Volume I - Executive Summary.

3. Ibid, Volume II - UPH Design Criteria

4. Ibid, Volume III - CAES Design Criteria.

5. Ibid, Volume IV - System Studies

6. Ibid, Volume V - Site Selection

7. Ibid, Volume VI - Site Investigation, Shallow Drilling.

8. Ibid, Volume VII - Site Investigation, Deep Drilling.

9. Ibid, Volume VIII - UPH Design Approaches.

10. Ibid, Volume VIII, Appendix A - Upper Reservoir.

11. Ibid, Volume VIII, Appendix B - Shafts.

12. Ibid, Volume VIII, Appendix C - Heavy Hoist.

13. Ibid, Volume VIII, Appendix D - Power Facilities.

14. Ibid, Volume VIII, Appendix E - Lower Reservoir.

15. Ibid, Volume IX, CAES Design Approaches.

16. Ibid, Volume IX, Appendix A - Air Storage System.

17. Ibid, Volume IX, Appendix B - Champagne Effect.

18. Ibid, Volume IX, Appendix C - Major Mechanical Equipment.

19. Ibid, Volume IX, Appendix D - Mechanical Systems.

20. Ibid, Volume IX, Appendix E - Electrical Systems.

21. Ibid, Volume X - Environmental Studies.

22. Ibid, Volume XI - UPH Plant Design.

23. Ibid, Volume XII - CAES Plant Design

24. Ibid, Volume XIII - Preliminary Licensing Documentation.

25. Driggs, C. L. and Robb, P. W., "Energy Storage for PEPCO - The Economic Feasibility of Compressed Air and Underground Pumped Hydroelectric Energy Storage on the PEPCO System," ASME Winter Annual Meeting, 1980.

26. Maass, P. and Stys, Z. S., "Operation Experience with Huntorf," American Power Conference, April, 1980.

TABLE 1

PEPCO CAES PLANT CHARACTERISTICS

Design Criteria	
Plant Size	Nominal 1000 MW
Number of Units	4
Storage Capacity	10 hrs @ full load
Air Flow Generating/Compression	1:1
Fuel	ASTM No. 2 Fuel Oil
Major Equipment Design	
High Pressure Turbine	7 stage, 5:1 ratio
Low Pressure Turbine	5 stage, 11:1 ratio
High Pressure Compressor	4 stage, Centrifugal
Intermediate Pressure Compressor	4 stage, Centrifugal
Low Pressure Compressor	11 stage, Axial
Motor/Generator	Double Ended, H_2 cooled
Recuperator Effectiveness	80%
Air Storage System Design	
Live Storage Volume	720,000 yd^3
Total Cavern Volume	876,000 yd^3
Depth to Caverns	2400 feet
Air Shaft	4 ft ID/Steel Liner
Water Shaft	13 ft ID/Concrete Liner
Reservoir Dike Height	35 feet
Reservoir Volume	675 acre-feet
Performance	
Generation Capacity	231 MW/Unit
Compression Load	170 MW/Unit
Effective Heat Rate (10,000 BTU/Kwhr input)	11,400 BTU/KWhr
Fuel Input	4015 BTU/KWhr
Elec. Energy Ratio (KWhr-In/KWhr-Out)	0.736
Cost & Construction Schedule	
Cost, incl. 45% indirects, w/o IDC	$550/KW, 1979$
Licensing Period	4 years
Construction Period	5½ years

POTOMAC ELECTRIC POWER COMPANY

UNDERGROUND PUMPED HYDRO STUDY

PETER E. SCHAUB
Potomac Electric Power Company
Washington, D. C.

FOSTER PELTON
Acres American, Inc.
Buffalo, N. Y.

ABSTRACT

In September of 1977, Potomac Electric Power Company (PEPCO) contracted with the Department of Energy (DOE) and the Electric Power Research Institute (EPRI) for a preliminary design investigation of Underground Pumped Hydro (UPH) and Compressed Air Energy Storage (CAES). As a subcontractor to PEPCO, Acres American, Inc. performed the engineering and managed a team of specialist contractors.

The Final Report of this study is currently being prepared, and will be issued by EPRI. This report is contained in a total of thirteen volumes and ten appendices.

The paper provides only a cursory review of the key issues involved. UPH is briefly defined and major plant characteristics are stated.

BACKGROUND

This common topic is covered in a companion paper: "Potomac Electric Power Company Compressed Air Energy Storage Study."

THE "CAUPH" STUDY

The work undertaken for DOE/EPRI was designated The "CAUPH" Study for Compressed Air/Underground Pumped Hydro. The common history, objective, and plan of the study are discussed in a companion paper: "Potomac Electric Power Company Compressed Air Energy Storage Study."

Key Issues Relating to UPH

During preparation of the proposal for study to ERDA, certain key issues were identified as requiring investigation or development. Primary among these were the design of the shafts, storage caverns and powerhouse chambers. It was seen necessary to develop real data on the host rock body and then optimize construction cost and schedule on the basis of excavation techniques capable of yielding stable excavations of sufficient size and volume.

Establishing the plan operating head was seen also as a key issue. It was recognized that the head would be limited by the technology of pump-turbine machinery. A plan to evaluate machinery availability worldwide was developed.

A final issue identified in the proposal was the ability to design, construct and operate a hoist capable of lowering the generating equipment several thousand feet to the underground powerhouse. Since all underground access would be through vertical shafts, loads such as transformers weighing over 100 tons would have to be handled by the hoist.

THE UPH DESIGN

What is UPH?

A schematic of the Underground Pumped Hydro scheme is shown in Figure 1. During generation, water is released through a penstock shaft to the powerhouse elevation. The flow is split in horizontal tunnels and routed to the hydro turbine-generators. The turbine discharge into a series of underground caverns. An air vent shaft is connected to the caverns to keep them at atomospheric pressure. This assures maximum work from the water head. During pumping, off-peak, the turbine-generators are reversed to operate as motor-pumps. The storage caverns are at a higher elevation than the powerhouse in order to provide positive suction head on the pumps and avoid cavitation-erosion on the runners.

No fuel is consumed directly by UPH, and this makes it virtually benign environmentally. If pumping is accomplished with nuclear and coal-generated electricity, then peaking can be accomplished without the use of oil. UPH is about 70% efficient; that is, 140% of the energy generated must be supplied as pumping power. As long as the peaking power alternatives to UPH are more costly to operate than coal or nuclear units by at least 40%, UPH should have an operating cost advantage. If this advantage is sufficient to compensate for the higher con-

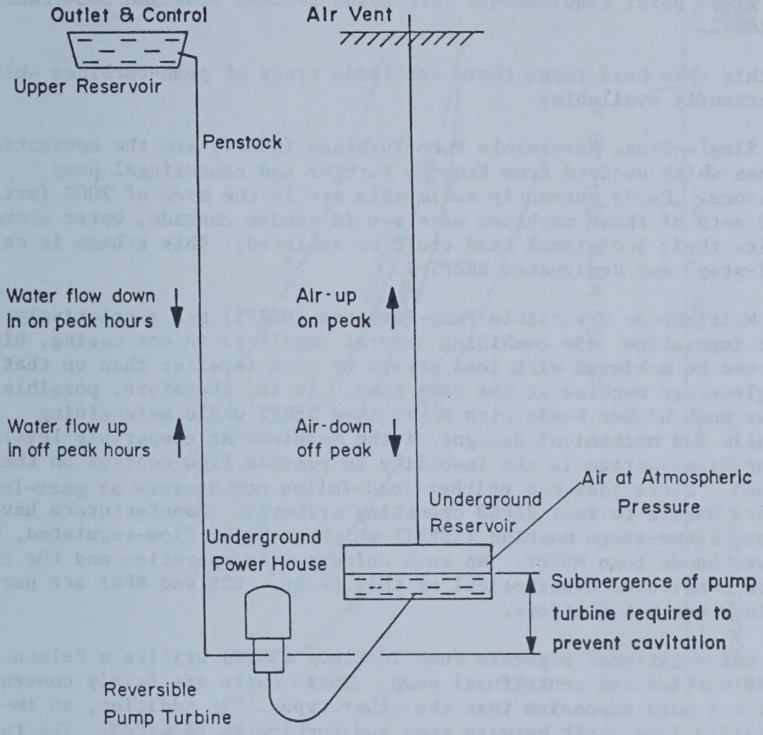

Outlet & Control
Upper Reservoir
Air Vent
Penstock
Water flow down
In on peak hours
Air-up
on peak
Water flow up
In off peak hours
Air-down
off peak
Air at Atmospheric
Pressure
Underground
Reservoir
Underground
Power House
Submergence of pump
turbine required to
prevent cavitation
Reversible
Pump Turbine

FIGURE 1 - UNDERGROUND PUMPED HYDRO SCHEMATIC

struction costs, UPH is an option worth pursuing.

Establishing Station Operating Head

The power available from UPH is a direct function of the product of head times flow. Since excavation of the storage caverns involves over 30% of total plant direct costs, their volume - and the water flow - should be minimized. This is accomplished by increasing the station head for a given power requirement. UPH heads between 2000 and 5000 feet are envisioned.

Within this head range there are three types of pump-turbines which are currently available:

- Single-Stage Reversible Pump-Turbines (SSRPT) are the conventional machines which evolved from Francis turbine and centrifugal pump experience. Heads currently achievable are in the area of 2000 feet. If two sets of these machines were set in tandem cascade, water storage at twice their individual head could be achieved. This scheme is called the "2-step" and designated SSRPT-2.

- Multi-Stage Reversible Pump-Turbines (MSRPT) are a relatively recent innovation. By combining several impellers in one casing, high heads can be achieved with less stress on each impeller than on that of a single-stage machine at the same head. It is, therefore, possible to achieve much higher heads with MSRPT than SSRPT while maintaining hydraulic and mechanical designs of the machines at comparable levels. A major disadvantage is the inability to provide flow control on these machines. Since they can neither load-follow nor operate at part-load, they are unable to meet PEPCO operating criteria. Manufacturers have proposed a two-stage machine (TSRPT) which could be flow-regulated, but at lower heads than MSRPT. No such units are in operation and the concept must be considered hypothetical at this point. DOE and EPRI are pursuing this in a new R & D effort.

- Uni-rotational Separate Pump-Turbines (USPT) utilize a Pelton impulse turbine and centrifugal pump. These units are fairly conventional but are more expensive than the other types. In addition, an impractically long shaft between pump and turbine is required. The turbine must operate above cavern water level and the pump must have a large submergence. Options such as a booster pump or separate motor and generator add substantially to the cost.

Figure 2 represents the culmination of Acres' review of pump-turbine alternatives. Characteristic cost curves were developed for the head range achievable by each type. The selection of head was made by lowest cost for available machinery meeting PEPCO operating requirements. The SSRPT-2 arrangement using equal drops of 2500 feet to 5000-foot depth was selected. Reference number (9) contains this analysis, including a thorough review of worldwide machinery experience. UPH pump-turbine equipment is reviewed also in reference number (26).

Underground Civil Design

In Task 2, several shallow borings were made into the body of rock selected during the station siting process. An attempt was made to

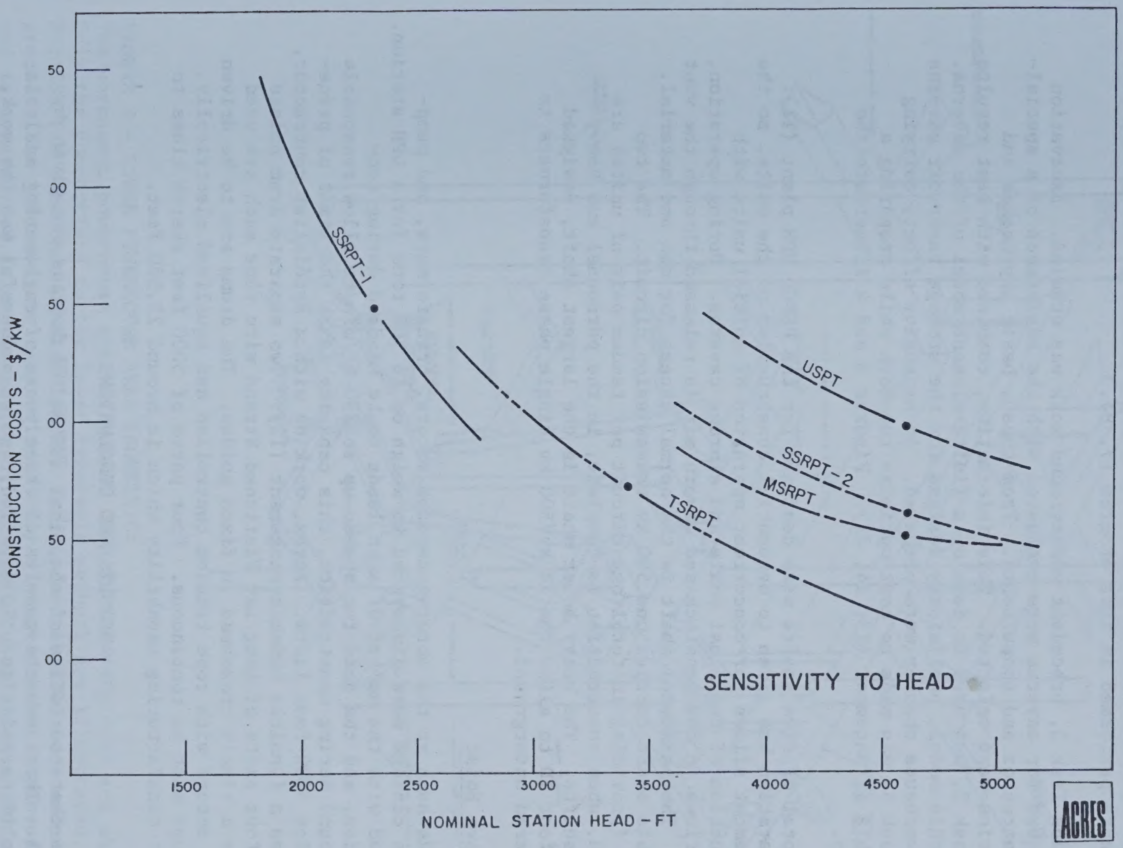


FIGURE 2 - UPH COST SENSITIVITY TO HEAD: Curves are characteristic for the pump-turbine arrangements shown. Costs are relative, and are based on the direct cost of a 2000MW, 10 hour storage plant.

complete a fully cored hole to 5000 feet but this was not successful. Severe anisotrophy of the rock - a high foliated gneiss - pulled the drill a total of 53°off vertical. At a depth of about 2500 feet and 3300 feet along the borehole, torque limits on the rods stopped the drilling. The UPH design was completed, nonetheless, with geological interpretation of drilling information to the full 5000-foot depth. Core strength and in-situ stress were among a series of laboratory and site tests performed in this program (7, 8).

During Task 3, precedent underground work was studied. Excavation techniques for caverns were evaluated with the assistance of a specialist contractor and consultants. From these, basic approaches and geometries were selected. This information, combined with test results from Task 2, was used to develop a finite element model of the caverns. Using this model, preliminary designs for the storage reservoir caverns and powerhouse chamber were prepared. An extensive effort, carrying into Task 5, was made to cost optimize the work while preparing a buildable arrangement (13, 14, 22). Figures 3 and 4 illustrate the results.

A total of five shafts were designed for the PEPCO UPH plant (11). Consideration was given to sequential construction of the units, so the arrangement allows for concurrent operation of initial units with construction of the final units and storage caverns. During operation, water flows in the penstock and cavern air is released through the vent shaft. The personnel shaft is the normal access for men and material. The cable shaft carries one 500 kv transmission circuit. The two others (providing in total one circuit per tandem pair of units) are carried, when construction is complete, in the personnel and heavy hoist shafts. The heavy hoist shaft is the largest shaft, designed at 30-foot ID to allow the 18 kv/500 kv single phase transformers to be lowered underground.

The Heavy Hoist

Loads such as the motor-generator rotors, transformers, and pump-turbine casings were discovered to weigh up to 300 tons for a UPH station. Combined with the number of such loads to be handled during construction, and the need for speeds up to 250 ft/min to allow reasonable turnaround during construction, this capacity is at the limit of precedence for 5000-foot lifts. Acres, working with a specialist contractor, designed a feasible hoist arrangement (12). Two separate drum hoists using four parts of Long Lay Flattened Strand wire rope each are used to move a single crosshead in fixed guides. The drums are to be driven by DC, motors with rope tension controlled and equalized electrically. Each rope must be continuous. Four parts of 5000 feet sketch close to current manufacturing capability which is around 22,500 feet.

RESULTS AND CONCLUSIONS

Some characteristics of the final PEPCO UPH designs are given in Table 1. These are the results of three years of engineering analysis and preliminary design. So detailed, and so successful was the work, that UPH is a technology which can now be implemented with confidence by U. S. utilities.

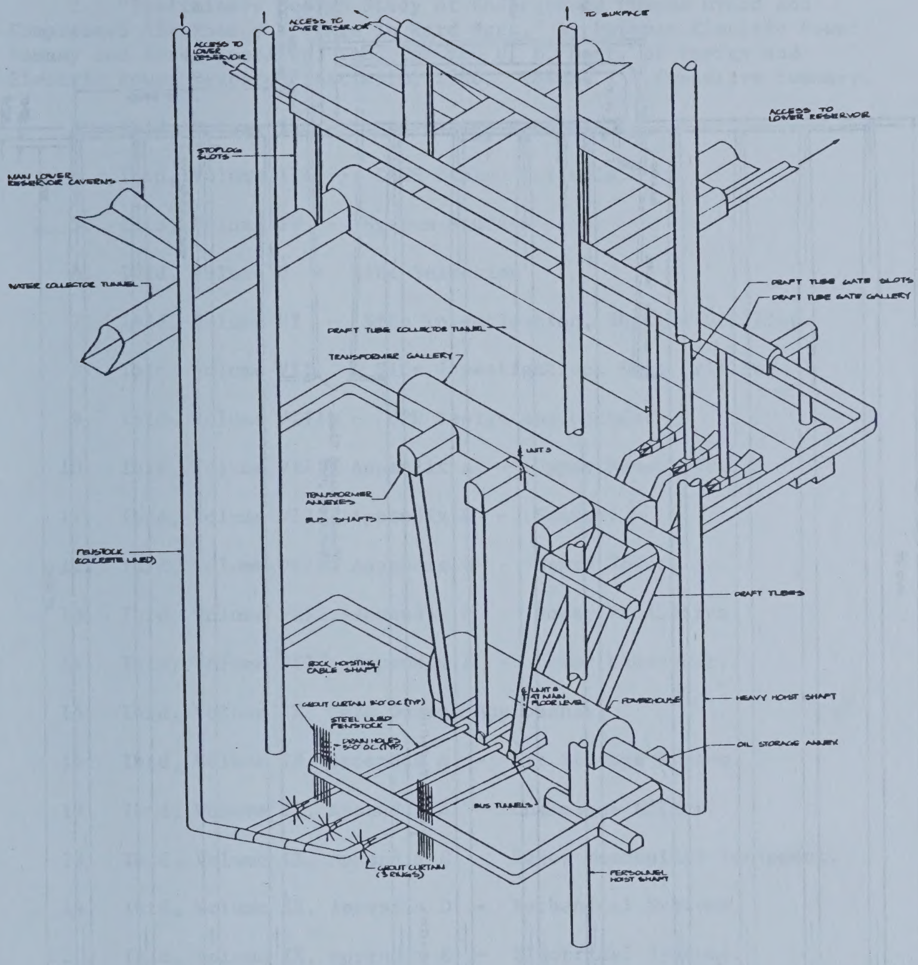

TO SURFACE
ACCESS TO LOWER RESERVOIR
ACCESS TO LOWER RESERVOIR
ACCESS TO LOWER RESERVOIR
STOPLOG SLOTS
MAIN LOWER RESERVOIR CAVERNS
WATER COLLECTOR TUNNEL
DRAFT TUBE GATE SLOTS
DRAFT TUBE GATE GALLERY
DRAFT TUBE COLLECTOR TUNNEL
TRANSFORMER GALLERY
UNIT 5
TRANSFORMER ANNEXES
BUS SHAFTS
PENSTOCK (CONCRETE LINED)
DRAFT TUBES
ROCK HOISTING / CABLE SHAFT
GROUT CURTAIN 10'0" O.C. (TYP)
UNIT 6 AT MAIN FLOOR LEVEL
POWERHOUSE
HEAVY HOIST SHAFT
STEEL LINED PENSTOCK
DRAIN HOLES 5'0" O.C. (TYP)
OIL STORAGE ANNEX
BUS TUNNELS
GROUT CURTAIN (3 RINGS)
PERSONNEL HOIST SHAFT

FIGURE 3 - LOWER POWERHOUSE AREA ISOMETRIC:
The permanent powerhouse excavations at 5000 feet underground are shown. Only the connections to the lower reservoir network are illustrated. For SSRPT-2 a second, similar complex is required at 2500 feet underground. The intermediate reservoir is only a surge chamber, however, and is constructed as an elongation of the Draft Tube Collector Tunnel.

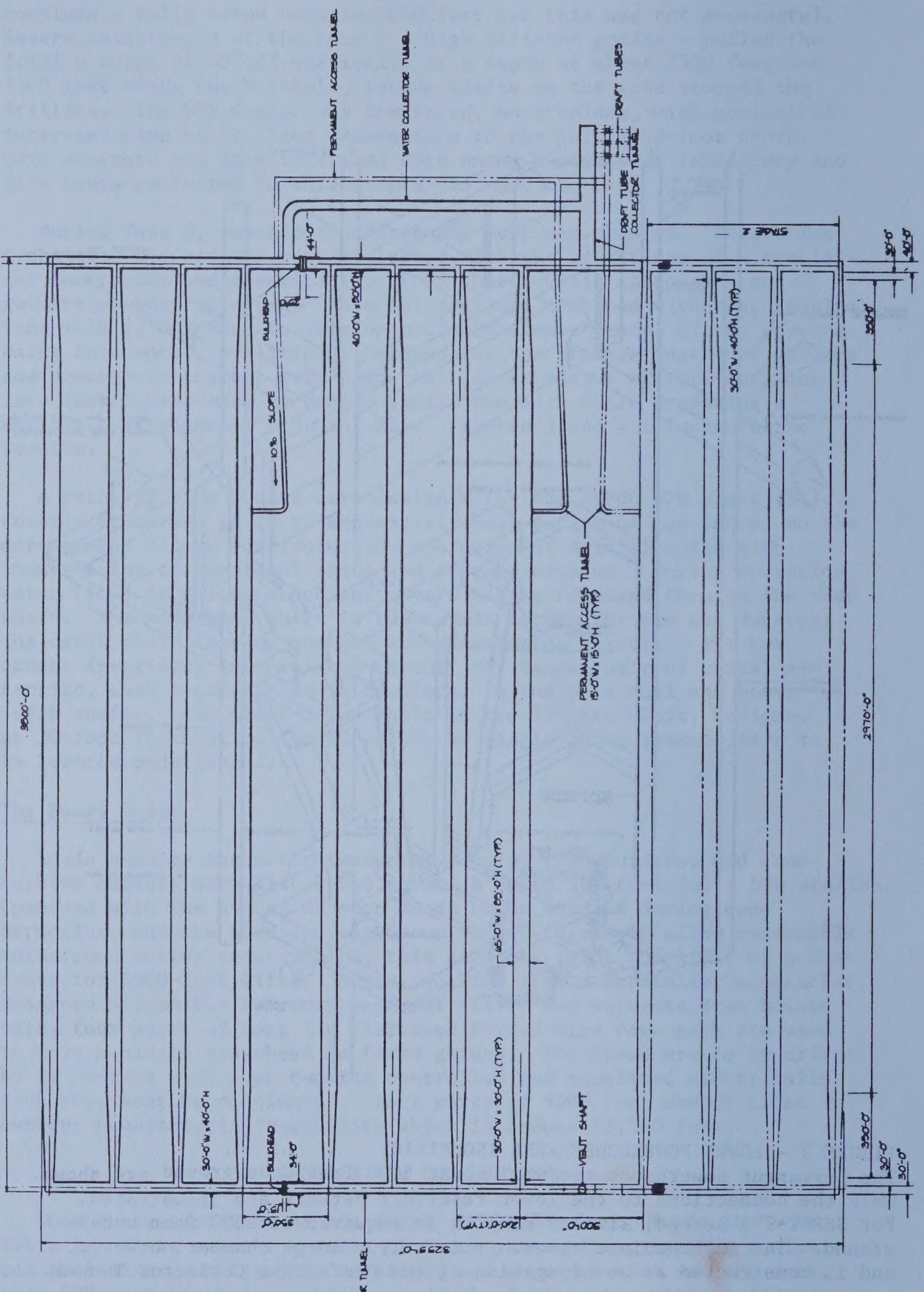

PERMANENT ACCESS TUNNEL
WATER COLLECTOR TUNNEL
DRAFT TUBES
DRAFT TUBE COLLECTOR TUNNEL
STAGE 2
30'-0"
40'-0"
44'-0"
BULKHEAD
15'-0"
40'-0"W x 50'-0"H
30'-0"W x 40'-0"H (TYP)
350'-0"
10% SLOPE
3600'-0"
2970'-0"
PERMANENT ACCESS TUNNEL 15'-0"W x 15'-0"H (TYP)
65'-0"W x 85'-0"H (TYP)
30'-0"W x 30'-0"H (TYP)
30'-0"W x 40'-0"H
BULKHEAD
34'-0"
VENT SHAFT
350'-0"
30'-0"
30'-0"
15'-0"
350'-0"
200'-0"(TYP)
300'-0"
3235'-0"
AIR COLLECTOR TUNNEL

FIGURE 4 - PLAN VIEW, UPH LOWER RESERVOIR

REFERENCES

1. Dickinson, W. H., "Muddy Run pumped storage," Electrical World, July 4, 1966, pp. 31-33.

2. "Preliminary Design Study of Underground Pumped Hydro and Compressed Air Energy Storage in Hard Rock," by Potomac Electric Power Comany and Acres American, Inc. for the U. S. Dept. of Energy and Electric Power Research Institute, 1980. Volume I - Executive Summary.

3. Ibid, Volume II - UPH Design Criteria

4. Ibid, Volume III - CAES Design Criteria.

5. Ibid, Volume IV - System Studies

6. Ibid, Volume V - Site Selection

7. Ibid, Volume VI - Site Investigation, Shallow Drilling.

8. Ibid, Volume VII - Site Investigation, Deep Drilling.

9. Ibid, Volume VIII - UPH Design Approaches.

10. Ibid, Volume VIII, Appendix A - Upper Reservoir.

11. Ibid, Volume VIII, Appendix B - Shafts.

12. Ibid, Volume VIII, Appendix C - Heavy Hoist.

13. Ibid, Volume VIII, Appendix D - Power Facilities.

14. Ibid, Volume VIII, Appendix E - Lower Reservoir.

15. Ibid, Volume IX, CAES Design Approaches.

16. Ibid, Volume IX, Appendix A - Air Storage System.

17. Ibid, Volume IX, Appendix B - Champagne Effect.

18. Ibid, Volume IX, Appendix C - Major Mechanical Equipment.

19. Ibid, Volume IX, Appendix D - Mechanical Systems.

20. Ibid, Volume IX, Appendix E - Electrical Systems.

21. Ibid, Volume X - Environmental Studies.

22. Ibid, Volume XI - UPH Plant Design.

23. Ibid, Volume XII - CAES Plant Design

24. Ibid, Volume XIII - Preliminary Licensing Documentation.

25. Driggs, C. L. and Robb, P. W., "Energy Storage for PEPCO - The Economic Feasibility of Compressed Air and Underground Pumped Hydroelectric Energy Storage on the PEPCO System," ASME Winter Annual Meeting, 1980.

26. Rodrique, P., "The Selection of High Head Pump-Turbine Equipment for Underground Pumped Hydro Energy Storage Application," paper, Pump-Turbine Symposium (ASME publication No. G00144), Joint ASME-ASCE Conference, Niagara Falls, June, 1979.

TABLE 1

PEPCO UPH PLANT CHARACTERISTICS

Design Criteria	
Plant Size	Nominal 2000 MW
Storage Capacity	10 hrs @ full load
Selected Design Configuration	
Arrangement	Two-step cascade
Pump-Turbine	Single-stage reversible
Number of Units	6: 3 ea @ two levels
Speed	720 rpm
Head	2500 feet per level
Motor-Generator	370 MVA, water-cooled
Hydraulic Design	
Upper Reservoir Height/Capacity	70 ft/4965 acre-ft
Construction	Compacted rockfill, asphalt lined
Penstock, Vertical	19 ft ID, concrete lined
Intermediate Reservoir	Tunnel, 85 ft H X 65 ft W X 3900 ft L
Volume	602,000 yd^3
Lower Reservoir	12 tunnels 85 ft H X 65 ft W X 3800 ft L
Volume	7,860,000 yd^3
Shafts and Powerhouses	
Vent Shaft	18 ft ID, concrete lined
Personnel Shaft	23 ft ID, concrete lined
Cable Shaft	24 ft ID, concrete lined
Heavy Hoist Shaft	30 ft ID, concrete lined
Powerhouses	44 ft H X 75 ft W X 313 ft L
Performance	
Hours/day Generating/unit	10 @ 340 MW
Hours/day Pumping/unit	12 @ 345 MW
Kwhr-out/Kwhr-in	78.2%
Daily Start-Up Losses	1%
Net Plant Efficiency	77.2%
Cost & Construction Schedule	
Cost, incl. 45% indirects w/o IDC	$605/kw, 1979 $
Licensing Period	4 years
Construction Period (first two units)	8 years

SIMULATION AND OPERATIONS OF LARGE WIND MACHINES

O. WASYNCZUK
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

The operating characteristics of a class of large, multi-megawatt wind turbine generators is examined. Design features relevant to the dynamic and transient behavior of this particular class of wind turbine generators are described. Results from hybrid computer simulation studies are included to illustrate the dynamic behavior of this class of wind turbine generators operating in a turbulent wind environment and also the transient behavior following electrical disturbances.

I. INTRODUCTION

The behavior of a wind turbine generator (WTG) when connected to a power system is an area of research and investigation. Of concern is the dynamic behavior of WTGs operating in a turbulent wind environment and also the transient behavior following electrical system disturbances. Common to the design of WTGs of the multi-megawatt size is a means of decoupling the wind fluctuations from the electrical system and the electrical disturbances from the blades. This can be accomplished by a soft, low speed shaft or by a spring mounted gearbox which yields similar mechanical characteristics as the soft shaft. In [1], it is illustrated that once a WTG of this type is disconnected from the power system following an electrical disturbance, it is impossible to maintain synchronism. The effects of the soft shaft on the operation of a WTG during normal, turbulent wind fluctuations are set forth in [2].

In this paper, the operating characteristics of a large WTG is investigated. A generic WTG is described and methods of simulation are discussed. Results from a hybrid computer simulation study are used to illustrate the dynamic behavior of the WTG operating in a turbulent wind environment and the transient behavior following electrical disturbances.

II. WTG DESCRIPTION AND MODELING

The WTG described is generically similar to the proposed MOD-2 which is representative of a class of WTG's of the multi-megawatt size to be synchronously connected into utility grids. A brief, general description is given here. A detailed description of the MOD-2 design is given in [1-5].

The WTG studied has a blade speed of 17.55 rpm geared to a 4-pole synchronous generator. The generator is rated 3.125 MVA at 0.8 power factor with a 3 phase, 60 Hz, 4.16 kV output. The exciter is a brushless rotating bridge type with the armature windings mounted on the rotor and the field windings mounted on the stator. The armature voltages of the exciter are rectified by a 6 pulse, 3 phase bridge rectifier and supplied directly to the field of the generator. The exciter field voltage is provided by a solid state voltage regulator which monitors the generator terminal voltage. The electrical system block diagram is shown in Fig. 1.

The blades of the WTG are an upwind horizontal axis propeller type, each 150 feet long with 45 feet of controllable tip length. For wind velocities below cut-in velocity (13 mph) or above cut-out velocity (45 mph), the blades are feathered and power is not extracted from the wind. Power production increases between cut-in and rated power output (low mode) as the wind velocity ranges from 13 to 28 mph. Rated power is produced as the velocity ranges from 28 to 45 mph, the cut-out velocity. In the low mode, the blade pitch angle is held fixed so as to maximize blade power output. In the high mode, the blade pitch angle is controlled so as to regulate blade power output. The pitch control in the high mode utilizes the difference between measured and rated power output and the error in measured hub speed. A simplified block diagram of the pitch control is given in Fig. 2.

The blade torque is transmitted to the generator via the drive train. Included in the drive train are the hub, quill shaft, gearbox (1:102.5), high speed shaft and generator rotor. The simplified diagram of the mechanical system is shown in Fig. 3. In Fig. 3, the inertia of the low speed shaft is distributed between the hub and gearbox. The high speed shaft is assumed to be rigid whereupon the inertia of the high speed shaft is included with the generator and gearbox as a single inertia. The wind torque on blade 1 is denoted T_{wB1} and T_E is the electromagnetic (generator) torque; all other variables are self explanatory. The simplified diagram of Fig. 3 is helpful in interpreting the results of computer studies given later. Data for the WTG (MOD-2 data) are given in Appendix A.

The wind torque on each blade, T_{wB}, can be expressed as

$$T_{wB} = \frac{1}{2} \rho A V_w^2 C_p \Gamma \tag{1}$$

where ρ is the density of the air, V_w is the wind velocity and Γ is the tip speed ratio defined as

$$\Gamma = \frac{V_w}{\omega_B} \tag{2}$$

where ω_B is the angular velocity of the blade. The power coefficient, C_p, is a function of the tip speed ratio and the blade pitch angle. This functional relationship is illustrated in Fig. 4 for the WTG studied.

Modeling techniques used to simulate the WTG are given in [1,2,4,5]. Briefly, the mechanical system is simulated as a lumped mass system by employing classical rotational dynamics. The generator is represented by Park's equations [6] with stator and network transients included. The exciter is a 3 phase synchronous machine which can also be represented using Park's equations. In [4], it was found unnecessary to represent the exciter in this amount of detail; instead, an IEEE Type 1 excitation system representation can be employed to model the effects of the exciter under transient conditions.

To portray the dynamic behavior of the WTG operating in a turbulent wind environment, a random wind model must be employed. In many studies where it has been important to determine the response of various structures to turbulent wind fluctuations, the approach has been to model the fluctuations in one spatial direction only and with spatial coherence assumed, i.e., with the wind speed in an area normal to the direction of flow represented as a function only of time. In this case, the wind fluctuations may be modeled as a single dimensional random process. In most instances, this process is described by its power spectral density. The theory of turbulent flow provides a mechanism for determining the general form of the power density spectrum and emperical measurements may be used to determine the wind model parameters. In the computer studies given later, a wind model based on the Davenport power density spectrum is used [7]. Convenient methods of simulating a random process with a specified spectral density are given in [8].

This representation of wind turbulence is an approximation. The wind is more accurately represented as a multidimensional and multivariate random process since the wind variations have components in all spatial directions. In addition, the various components are functions of both time and spatial coordinates. This type of representation requires a more detailed aerodynamic blade model to map the multidimensional, multivariate process into the hub torque which is a function only of time.

The assumption of spatial coherence greatly simplifies simulation and analysis. Although this assumption is sufficient for low frequency variations caused by large eddies encompassing the swept area of the blade, it may tend to overemphasize higher frequency variations caused by smaller eddies encompassing only parts of the blade. Results obtained from a study where real time wind data were used to predict the dynamic performance of a generically similar WTG [9] give indication that this type of wind model is adequate for predicting lower frequency variations. The effects of spatial wind variations on the dynamic behavior of large WTGs is an area of research and investigation.

III. DYNAMIC BEHAVIOR

In this study, a detailed hybrid computer simulation is used to illustrate the dynamic behavior of the WTG during random wind variations. The dynamic response of the WTG differs depending upon the mode of operation of the pitch control. In this study, operation in low and high modes is considered; however, the dynamic behavior of the system during switching between the modes is beyond the scope of this paper.

Low Mode Behavior

In the following study it is assumed that a single WTG is connected to a large system (infinite bus) through a generator step up transformer and a transmission line (Fig. 5). The per unit impedance of the generator step up transformer is 0.007+j0.05 on a 3.125 MVA base. The transmission line reactance is X_L =0.1 pu with R_L = 0.10 X_L.

In order to examine the dynamic behavior of the WTG system operating in a turbulent wind environment, a random wind model based on the Davenport spectrum is used. In the first case, it is assumed that the mean wind speed is 20 mph and that the standard deviation of the turbulent fluctuations is 2.3 mph. A representative computer generated response is shown in Fig. 6 where the following variables are plotted V_W - wind speed in mph; T_{BH} - blade to hub torque in pu on a 3.125 MVA base; T_{HG} - hub to generator (quill shaft) torque in pu on a 3.125 MVA base; P_E - electric power in MW; δ - machine power angle relative to the infinite bus; V_T - machine terminal voltage magnitude in per unit; T_{WB} - total developed blade torque (blades 1 and 2) in pu on a 3.125 MVA base.

The generated electric power is shown in Fig. 6; the magnitude of the fluctuating component is of the same order as the mean power (1.0 MW). These fluctuations will not have a significant impact provided that the level of wind generation is small relative to the installed system capacity and that the interconnections are into a strong grid. The large oscillatory fluctuations of approximately 0.9 rad/sec (0.14 Hz) which appear in the blade to hub torque, the hub to generator (quill) shaft torque and the machine power angle are excited by the wind. These variations occur even through the variations in the developed blade torque, T_{WB}, follow the relatively small fluctuations in wind speed.

The presence of these large oscillatory fluctuations can be explained from a torsional analysis of the drive train. The details of such an analysis are given in [2] where it is shown that the frequency of the oscillations coincides with a torsional mode where the hub and blades swing against the generator through the quill shaft (soft shaft mode). For the data given in the Appendix, the frequency of this mode of oscillation is 0.85 rad/sec and is very lightly damped. As a consequence of the dynamic characteristics of the drive train, small variations in the developed blade torque, T_{wB}, caused by wind turbulence at or near this frequency will be amplified by approximately 25 dB producing large oscillatory variations in the electric power as observed in Fig. 6.

High Mode Behavior

With wind speeds above 28 mph and a pitch angle of zero degrees, the WTG will produce above rated power. Consequently, the mode of control is changed from the fixed pitch mode (low mode) to the variable pitch mode (high mode) where the electric power is regulated by the pitch control which uses proportional plus integral power and proportional hub speed feedback signals (Fig. 2). In this study, the designed feedback gains were adjusted to provide optimal dynamic performance for the wind model employed.

The investigation of the dynamic performance of the WTG in the high mode of operation was conducted using the same system (Fig. 5) with a mean wind speed of 35 mph. In this case, it is assumed that the standard deviation of the turbulent fluctuations in wind speed is 3.9 mph. A representative time response plot is shown in Fig. 7 where the same variables are plotted as in Fig. 6. A significant improvement in power quality is obtained. In addition, the variations in blade to hub torque, hub to generator torque and machine power angle are reduced significantly. This is due to the feedback signals used to regulate the generated power. The speed feedback essentially adds damping to the soft shaft mode and the power feedback serves to oppose the effects of changes in wind speed due to turbulence.

IV. TRANSIENT BEHAVIOR

For the WTG studied, the quill shaft twists approximately 3.6° under full load. Therefore, during electrical disturbances (e.g., load rejection), the quill shaft will untwist. With a gear ratio of approximately 100:1, the untwisting of 3.6° on the low speed side corresponds to a rotation of 360° on the high speed side which is equivalent to 720° electrically since the generator is a 4-pole machine. The drive train is a dynamic system and during the first transient overswing, the rotor angle will approach 1300° electrically. The advance in rotor angle is so rapid that synchronism could not be maintained following most system disturbances with normal switching and de-ionization times.

This feature of the mechanical system is illustrated in Fig. 8. Initially, the WTG is operating at rated power. The circuit breaker connecting the WTG to the electric system is opened (load rejection) whereupon the synchronous machine appears only as a rotating mass to the mechanical system. The following variables are plotted in Fig. 8: T_{wB1} - torque on blade 1, $\Delta\omega_{B1}$ - percent change in speed of blade 1, T_{BH} - torque transmitted from blades to hub, $\Delta\omega_{H}$ - percent change in hub speed, T_{HG} - quill shaft torque, $\Delta\omega_{G}$ - percent change in generator speed, δ - rotor angle in electrical degrees, and T_{E} - electromagnetic torque. All torques are expressed in per unit with respect to rated output.

The influence of the soft shaft is demonstrated in the response of $\Delta\omega_G$, and δ. In approximately 0.9 sec after load rejection, the rotor angle has exceeded 1300 electrical degrees. This advance is so rapid that synchronous operation could not be maintained unless the generator is reconnected into the system within 0.160 sec following loss of load [1]. This critical switching time is far too short to permit the appropriate switching operations or fault de-ionization. Although the disturbance studied is load rejection, any electrical disturbance which results in a significant reduction in the electric power can cause a loss of synchronism before normal corrective actions can be taken. Once synchronism is lost, the recommended procedure is to stall the wind turbine, and proceed with normal start-up and synchronization; a procedure which may require from 10 to 20 minutes. Methods of resynchronization which markedly reduce the time to re-establish normal operating conditions are set forth in [10].

V. CONCLUSIONS

The operation of a large WTG connected to an electric grid was investigated. In this type of wind turbine, a soft, low speed shaft is used in the drive train. The presence of this soft shaft gives rise to a low frequency, lightly damped torsional mode involving an oscillation where the hub and blades swing against the generator through the soft shaft. In the fixed pitch mode of operation, this mode can be excited by normal wind variations resulting in large oscillatory fluctuations in the drive train torques as well as in the generated electric power. In the high, or variable pitch mode of operation, the proportional plus integral power and proportional hub speed feedback signals used to control the blade pitch angle can be suitably adjusted to provide satisfactory dynamic performance of the WTG operating in a turbulent wind environment.

The transient behavior of the WTG following electrical disturbances is characterized by a rapid untwisting of the soft, low speed shaft. This feature of the mechanical system is so overriding that in most cases synchronism could not be maintained by normal switching procedures, even if high speed reclosing were permitted. This problem of reliability may influence the connecting arrangement and configuration of wind farms as well as line and load switching procedures in the vicinity of wind turbine generators.

VI APPENDIX A: MOD-2 SYSTEM DATA

The mechanical system parameters referred to a 2.5 MVA base are

Inertias

$2J_{B1} = 18.23$ $2J_H = 0.9527$

$2J_{B2} = 18.23$ $2J_G = 2.091$

Shaft Stiffness

$\omega_b K_{B1H} = 861.1$ $\omega_b K_{HG} = 28.40$

$\omega_b K_{B2H} = 861.1$

Damping

$\omega_b D_{B1} = 0.00458$ $\omega_b d_{B1H} = 29.31$

$\omega_b D_{B2} = 0.00458$ $\omega_b d_{B2H} = 29.31$

$\omega_b D_H = 0.01108$ $\omega_b d_{HG} = 1.832$

$\omega_b D_G = 0.03610$

The per unit machine parameters on a 3.125 MVA base are

$R_s = 0.0075$ $R_{fd} = 0.001063$
$X_{\ell s} = 0.081$ $X_{\ell fd} = 0.1884$
$X_{mq} = 0.929$ $X_{md} = 1.85$
$R_{kq} = 0.00988$ $R_{kd} = 0.01423$
$X_{\ell kq} = 0.06643$ $X_{\ell kd} = 0.1049$

ACKNOWLEDGEMENTS

This work was done under DOE Contract No. DE-AS02-77ET 29 100 entitled "Security Assessment of Power Systems with the Integration of Wind Energy." The data for the MOD-2 were obtained from NASA/Lewis Research Center, Cleveland, Ohio.

LIST OF REFERENCES

[1] P. C. Krause and D. T. Man, "Transient Behavior of a Class of Wind Turbine Generators During Electrical Disturbances," Being reviewed by IEEE.

[2] O. Wasynczuk, D. T. Man and J. P. Sullivan, "Dynamic Behavior of a Class of Wind Turbine Generators During Random Wind Fluctuations," Being reviewed by IEEE.

[3] L. J. Gilbert and D. M. Triezenberg, "Lewis Research Center Studies of Multiple Large Wind Turbine Generators on a Utility Network," Workshop on Economic and Operational Requirements and Status of Large Scale Wind Systems, Monterey, CA, March 28-30, 1979, EPRI Report ER-1110-SR, July 1979, pp. 338-402.

[4] P. C. Krause, "Analysis and Simulation of Wind Energy System," NASA Final Report, NASA Grant NSG-3237, November 1979.

[5] D. T. Man, "Transient Behavior of a MOD-2 Wind Turbine Generator," Purdue University Technical Report, TR EE 80-22, May 1980.

[6] R. H. Park, "Two-Reaction Theory of Synchronous Machines - Generalized Method of Analysis - Part 1," AIEE Transactions, Vol. 48, July 1929, pp. 716-727.

[7] A. G. Davenport, "The Spectrum of Horizontal Gustiness Near the Ground in High Winds," Quarterly Journal of the Royal Meteorological Society, Vol. 87, April 1961, pp. 194-211.

[8] M. Shinozuka, C-M Jan, "Digital Simulation of Random Processes and Its Applications," Journal of Sound and Vibration, Vol. 25, November 1972, pp. 111-128.

[9] J. M. Kos, "On Line Control of a Large Horizontal Axis Wind Energy Conversion System and Its Performance in a Turbulent Wind Environment," Proceedings of the 13th Intersociety Energy Conversion Engineering Conference, August 1978, pp. 2064-2073.

[10] P. C. Krause and O. Wasynczuk, "Methods of Resynchronizing Wind Turbine Generators," Being reviewed by IEEE.

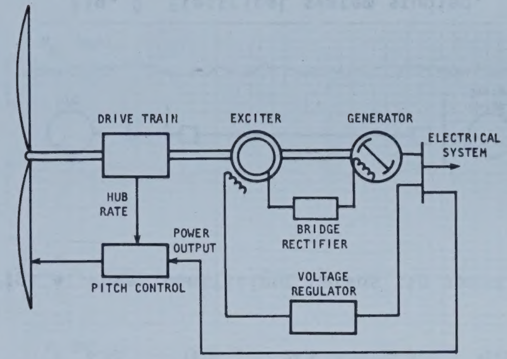

DRIVE TRAIN
EXCITER
GENERATOR
ELECTRICAL SYSTEM
HUB RATE
POWER OUTPUT
BRIDGE RECTIFIER
PITCH CONTROL
VOLTAGE REGULATOR

Fig. 1 Electrical system block diagram.

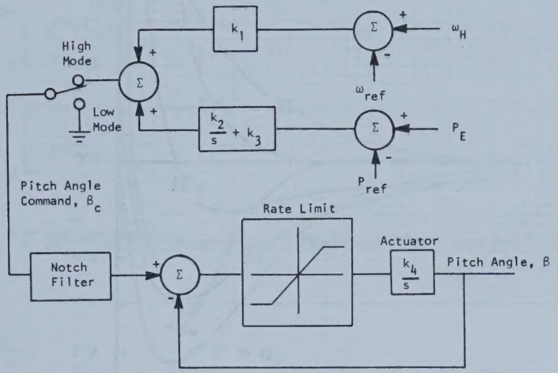

k_1
Σ
+
-
ω_H
ω_{ref}
High Mode
Low Mode
$\frac{k_2}{s} + k_3$
P_E
P_{ref}
Pitch Angle Command, β_c
Rate Limit
Actuator
$\frac{k_4}{s}$
Pitch Angle, β
Notch Filter

Fig. 2 Block diagram of pitch control.

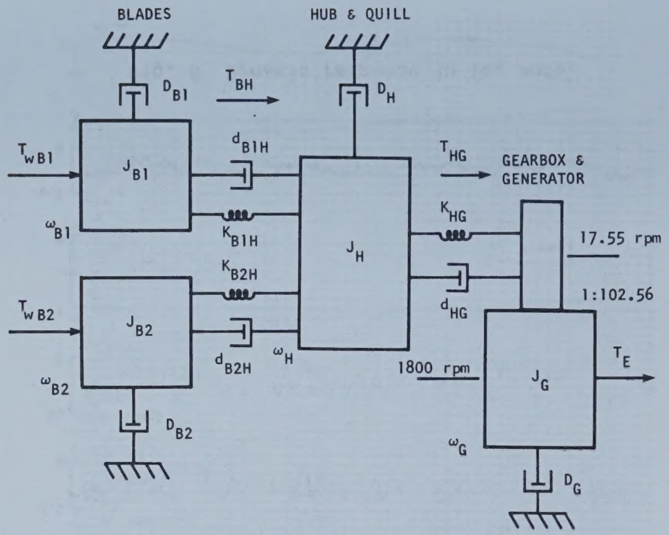

BLADES
HUB & QUILL
D_{B1}
T_{BH}
D_H
T_{wB1}
J_{B1}
d_{B1H}
T_{HG}
GEARBOX & GENERATOR
ω_{B1}
K_{B1H}
K_{HG}
J_H
17.55 rpm
K_{B2H}
1:102.56
T_{wB2}
J_{B2}
d_{HG}
d_{B2H}
ω_H
1800 rpm
T_E
J_G
ω_{B2}
D_{B2}
ω_G
D_G

Fig. 3 Mechanical system diagram.

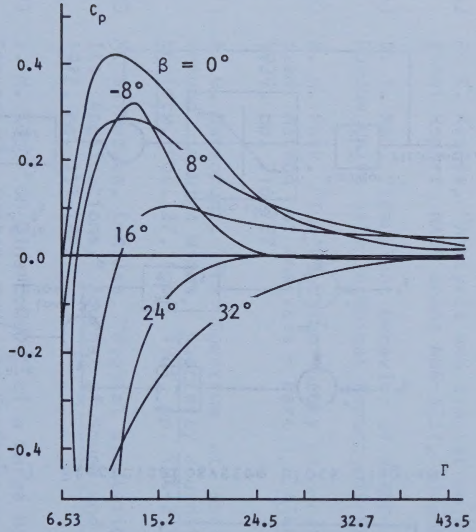


Fig. 4 Power coefficient versus tip speed ratio.

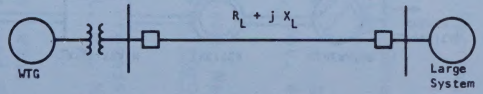

WTG
R_L + j X_L
Large System

Fig. 5 Electrical system studied.

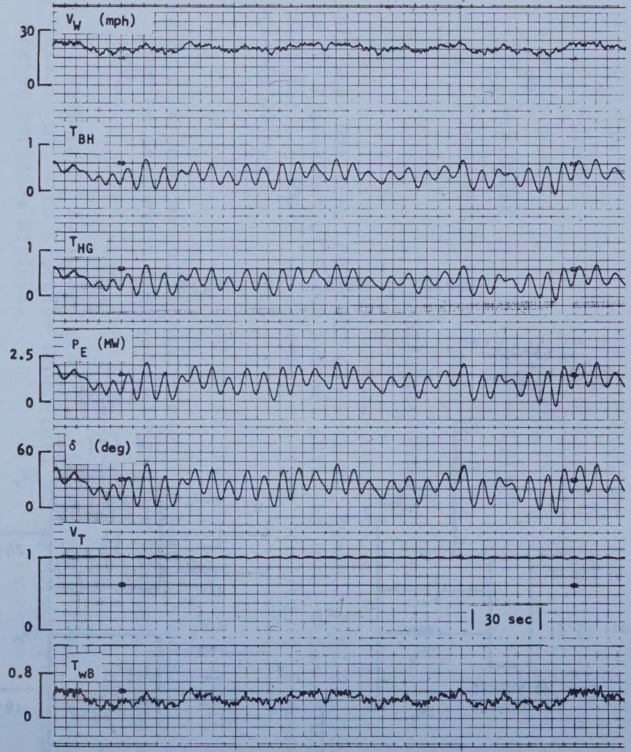


Fig. 6 Dynamic response in low mode.

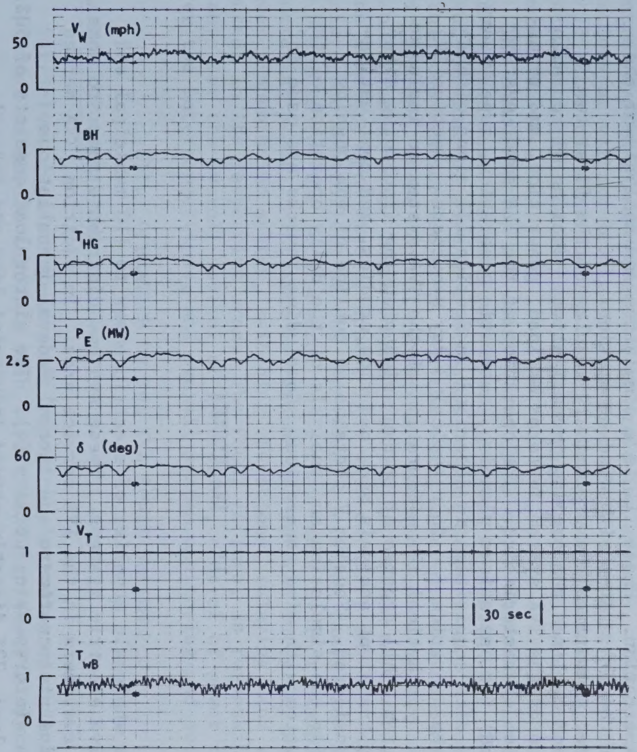


Fig. 7 Dynamic response in high mode.

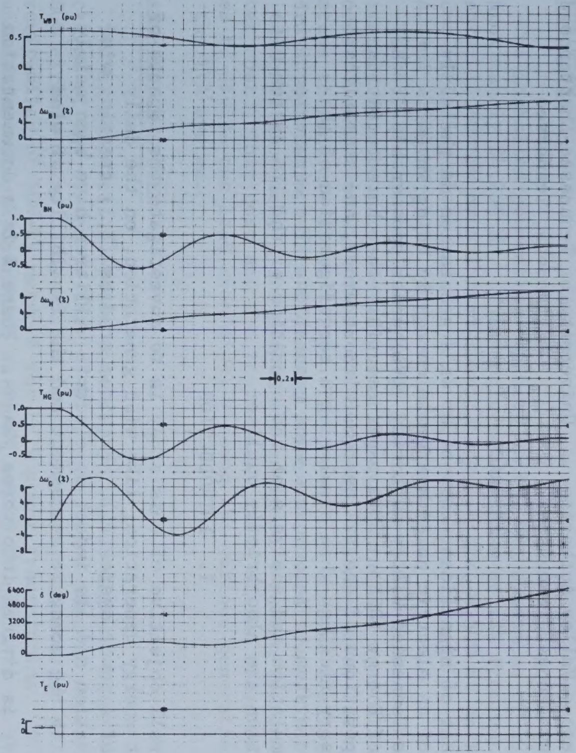


Fig. 8 Transient behavior following load rejection.

AN IMPACT ASSESSMENT OF MOD-2 WIND GENERATORS ON THE TVA POWER SYSTEM

D. W. HILSON N. D. SADANANDAN
Tennessee Valley Authority
Chattanooga, Tennessee

H. M. SENDAULA R. R. JOHNSON
University of Tennessee-Chattanooga
Chattanooga, Tennessee

ABSTRACT

The impact of significant amounts of wind generation on the daily operations of an electric power system is currently being investigated by the Tennessee Valley Authority (TVA) under sponsorship of the Department of Energy (DOE). This paper describes the characteristics and daily operations of the TVA system (without wind), and summarizes the preliminary findings when simulating a reconfiguration of the transmission system to accommodate wind generation. The ongoing activities for simulation and analysis of the operation of the TVA system with wind generation are also briefly outlined.

I. INTRODUCTION

The overall objective of this study is to investigate the impacts of intermittent energy generation, like wind generation, on the daily operations of a power system (1). Previous studies (2,3) treating wind energy generation and its impact on electric utilities have been mainly concerned with the economic assessment and the feasibility of wind systems.

The study is divided into three phases. Phase I is a documentation of the characteristics and the daily operations of the TVA system for typical days in each of the four seasons. Phase II is a hypothetical reconfiguration of the TVA system to include wind generation at various levels of penetration. Phase III is a simulation and analysis of the operation of the TVA system with wind generation.

This paper summarizes the preliminary results of the first two phases (4) and outlines the ongoing investigation. Section II contains a description of the TVA system in terms of load, generation, and transmission system characteristics. The operation of the TVA power system (without wind) for operations planning, automatic generation, supervisory control, and data acquisition is described in Section III along with a summary of the operation for four typical 3-day sequences. Section IV outlines the hypothetical reconfiguration of the TVA system to include wind generation. All aspects of transmission planning (including site development, transient analysis and load flow studies, and cost estimation) are considered. Section V contains an outline of the ongoing activities for the simulation and analysis of the daily operation of the TVA power system with wind generation. Concluding remarks are made in Section VI.

II. TVA SYSTEM

LOAD

TVA is a wholesale supplier of electric power to 110 municipal and 50 cooperative electric systems. These systems distribute power to some 2.5 million customers in the TVA service area. In addition, TVA serves 49 large industrial customers and several Federal installations.

TVA is a winter-peaking utility (4). The winter load consists of 50% residential load, 22% directly served industrial load and Federal establishments, and 28% commercial load. For the summer peak the load breakdown is approximately as follows: 26% directly served load, 37% residential, and 37% commercial load.

GENERATION

The installed generating capacity of the TVA system as of 1978 was approximately 28,250 MW. This included 3,450 MW or 12% from nuclear plants, 17,800 MW or 63% from fossil-fired plants, 4,500 MW or 16% from hydro including Alcoa and Corps of Engineers hydro plants operated by TVA, and 2,500 MW or 9% from combustion turbines.

TVA's hydro system is an energy-limited system and is operated primarily for the purposes of promoting navigation and controlling floods. Consistent with these two purposes, TVA also uses its hydro systems to generate electricity. The capacities and the loading of the hydro units are therefore dictated by the river conditions. For purposes of economic dispatch, hydro plants are assigned a pseudo-incremental water cost. This cost and the incremental rates of the steam units are used to divide the load economically among the various generating units on the TVA system.

TRANSMISSION SYSTEM

TVA's transmission system consists of 16,803 circuit-miles (4). The backbone of the system is the 161-kV transmission network consisting of 9,000 circuit-miles. The system is reinforced by an overlay of 1,700 circuit-miles of 500-kV network for bulk power transmission. A subtransmission network operating at 69 kV and 49 kV consists of approximately 5,760 miles.

The other components of the transmission system consist of a variety of subsystems at various voltage levels, switchyards, communication and control systems, and protective and relaying systems. TVA operates substations at voltage levels ranging from 500 kV for bulk power transmission to 13 kV at the distribution level. Various voltage and VAR control systems maintain the voltage levels within specified percentages of the nominal values.

The communication and control system is of an hierarchial structure. The highest level is the Power System Control Center (PSCC). The next level consists of five Area Dispatch Control Centers (ADCC) dispersed within the TVA service area. On the third level are the substations and switchyards. These control centers are interconnected by a variety of channel facilities. These channel facilities include microwave, power line carrier, open wire telephone line with carrier, cables, and VHF radio. The controllers at the individual operating facilities are responsible for the local operation of that facility (for example, at a generating plant). The ADCC is responsible for the integrity of the power transmission system, switching operations, relaying, and voltage control in its respective area. The PSCC, in turn, is responsible at all plants for the overall control of generation, control of power flowing over the interconnections, and overall system monitoring and security.

III. OPERATION OF THE TVA POWER SYSTEM

PLANNING OF OPERATIONS

Sufficient megawatt generating capacity, including operating reserves, is planned to ensure that for each future year for the TVA system the probability of failure to supply load shall not exceed the acceptable risk level (4). Load estimates, scheduled maintenance, hydrological conditions, and power available from neighboring utilities are used to determine the generating capabilities for the next 12 months.

Weekly planning of generation on the TVA system is mainly for scheduling hydro generation. A model of TVA's hydro system is used to determine such parameters as: average hydro generation, storage volumes, changes in storage volumes, accumulation of local inflows, plant discharges, maximum turbine discharge capability, estimated spills, and reservoir elevations.

The daily planning of operations consists of two parts. The first is unit commitment and the second is unit dispatch. Peak load estimates for the next day, operating reserve requirements, and unit availability data are used in committing units.

The unit dispatch program, also called the preschedule, attempts to determine the most economic distribution of generation among the various units. The results of the unit commitment procedure are inputs to this program. The program, utilizing incremental heat rates, pseudo-incremental water cost, fuel costs, reservoir elevations, and other data attempts to match the load with the available generation in the most economic manner.

AUTOMATIC GENERATION CONTROL

TVA, as one of the electric power producers forming a vast interconnected power grid in North America, must observe a number of rules of power system operations: (1) meet the internal load; (2) maintain scheduled interchange with its neighbors; and (3) share in the frequency regulation according to its capacity.

The primary system of TVA's automatic generation control (AGC) is a direct-digital control system consisting of a Xerox Sigma 5 process control computer at the Power System Control Center (PSCC). Analog systems and other small computers are used as backups. Load frequency control (LFC) and economic dispatch control (EDC) are implemented online using these computers. Tieline flows, system frequency, interchange schedules, frequency bias settings, and the frequency schedule are inputs to the load frequency control program. The control requirements are calculated every two seconds, and the outputs are transmitted to the generating plants to adjust system generation.

Every five minutes the computer executes the EDC program. The inputs are unit incremental costs, transmission loss coefficients, tieline flows, hydro plant generation readings, industrial loads, high and low limits of the generating units, unit capability readings, and plant availability data. The computer then determines the optimal operating point for each generating unit on control.

SUPERVISORY CONTROL

The five area dispatch control centers (ADCC) supervise control of major substations and a few hydro plants in their respective areas. Each ADCC master consists of a dual computer configuration with automatic switchover in case one computer fails and has an ultimate capacity to control and monitor a combination of 40 substations, hydro plants, and switchyards. Items controlled include circuit breakers, transformer tap changers, relay channel tests, and hydro unit controls. Data monitored includes breaker status, line amperes and voltages, transformer loading, hydro unit status, alarm conditions, and the capability for recording a sequence of events for a system disturbance.

Supervisory control and data acquisition (SCADA) terminals transmit data from remote stations for utilization at the ADCC and transmission on to the PSCC. Bidirectional information flow is continuously maintained between PSCC and ADCC over a variety of communication media.

EMERGENCY CONTROL

In order to prevent the cascading collapse of the system and systems interconnected with it when the frequency is changing radically, TVA has decided upon the following emergency actions:

(1) As the frequency declines below 59.95 Hz, all available resources will be utilized through automatic and non-automatic procedures.

(2) If frequency drops below 59.8 Hz, excess load will be dropped by implementing TVA's emergency load curtailment plan.

(3) Should the frequency drop below 59.5 Hz, preselected blocks of system load (about 40%) will be dropped with the operation of underfrequency relays.

(4) If the frequency drops below 59 Hz, additional load will be dropped manually by prearranged operating procedures. (Such loads will also be picked up manually as the system conditions permit.)

(5) Should the frequency drop below a predetermined frequency (presently 58 Hz) for a predetermined time (presently 1-1/2 minutes) system switching will be performed in accordance with a predetermined operating procedure to separate each generating plant from all parts of the system except those within which it is found able to restore to 60 Hz.

(6) When all efforts have failed, necessary switching will be performed to establish emergency routes for station service power from hydro plants to steam plants to permit orderly shutdown and prompt restarting of units.

OPERATIONS ON 12 TYPICAL DAYS

The system operations for the 12 days simulated represent a typical 3-day sequence in each season of 1978: spring, April 15, 16, and 17; summer, August 8, 9, and 10; fall, October 17, 18, and 19; and winter, December 12, 13, and 14.

Table 1 summarizes the operations during the critical hours on a selected day when the net system load or estimated generation deviated from the actual system load or actual generation by more than 500 MW (2-1/2%). Two types of deviations are considered: (1) net system deviations which occur in estimating the net system load and; (2) generation deviations which occur in estimating the generation capabilities of the system.

IV. CONFIGURATION OF THE TVA SYSTEM WITH WIND GENERATION

TRANSMISSION EXPANSION PLANNING

In this study the TVA transmission and subtransmission systems had to be appropriately modified hypothetically to accommodate wind energy conversion systems (WECS) generation for 5-30% penetration levels. Each unit is a MOD-2 represented by a synchronous generator having a rated output of 2.5 MW (5). At 5% penetration, 568 units and at 30% penetration 3,408 units would have to be connected to the system.

It is assumed that at any time 20% of the connected WECS capacity could go to zero over the next hour. All phases of planning were considered.

TVA's transmission planning criterion is as follows: The transmission of the system is planned and constructed so that more probable contingencies can be met without loss of load and the less probable contingencies can be sustained with some loss of load but without uncontrolled cascading interruptions. The system is designed so that during these contingencies the emergency rating of any facility is not exceeded.

CONNECTION OF WECS ON TRANSMISSION SYSTEM

The (WECS) generation units for 5-30% penetration were organized into farms of up to 24 units each. These farms were assumed to be connected through single 161/69-kV transformer banks into the 161-kV transmission network. All transmission for WECS units within a farm were at 69 kV over two separate lines.

The 69-kV lines, each with a power circuit breaker at the collector substation, were routed so as to limit loss of generation to 30 MW or 12 units. The minimum separation between MOD-2 individual unit installations was stipulated to be 10 rotor diameters (3,000 feet); hence, the 69-kV transmission line connecting adjacent WECS was 0.7 miles in length. The line impedance was 0.5% of that of the 4.16/69-kV unit step-up transformer bank. Because of the short length of line connecting adjacent WECS, the units connected to one collector line would therefore perform as if they were connected to a common point at their respective 69-kV unit step-up transformers. The method selected for connecting the WECS to the transmission system consisted of connecting any 24 dispersed MOD-2 units to two 69-kV lines (12 per line) which terminated at a collector substation, feeding into a 68 MVA, 161/69-kV transformer.

At 5% penetration there would be 23 such farms. At 30% penetration there would be 142 farms and about the same number of collector substations. Each collector substation was connected into the 161-kV grid by looping the existing 161-kV transmission line into the collector substation.

WECS TRANSIENT STABILITY ANALYSIS

The electrical and mechanical characteristics of the MOD-2 WECS are summarized in Table 2. Transient stability studies were made to determine whether the operating WECS units on a 69-kV collector line could be expected to survive a normally cleared fault on its collector line within the farm or on the 161-kV system. Figure 1 was used in this study.

The results indicate that the wind farm would be stable for 161-kV faults and reclosing 161-kV line breakers, provided the WECS were not separated from the system. Faults on a 69-kV collector line will cause instability as will be explained below in the example case studies.

Under full load, the turbine (soft) shaft of the MOD-2 WECS twists approximately 3.6° (7). For a 4-pole machine with a gear box ratio of 100:1 between the turbine shaft and the high speed shaft, there is normally over 700 electrical degrees of twist in the turbine shaft referred to the generator rotor. If the circuit breaker on the 69-kV line were to be opened, the untwisting of the soft shaft would cause, in less than one second, over 1,300 electrical degrees of twist in the turbine shaft and over 200 electrical degrees change in the rotor angle.

The following case studies have been selected as representative of the many transient stability studies performed during the investigation of the impact of WECS on the operation of the TVA power system. The circuit configuration for the following studies is shown in Figure 1 and the data assumed is given in Table 2.

Case 1--A 3-phase fault on the 161-kV line terminal (Location A) was cleared by 161-kV breakers 2 and 3 opening and locking out at 4-1/2 cycles (0.075 seconds). Prior to the fault, the Group 2 generator units were fully loaded at 98% power factor. Subgroups 1A to 1D were loaded to two-thirds, zero, one-third, and full load, respectively; and all were at 85% power factor. For this particular fault location and WECS loading conditions, the result is that the farm was stable as is evidenced by the curves shown in Figure 2.

Case 2--With the same prefault conditions and the same fault location as in Case 1, line breakers 2 and 3 tripped at 4-1/2 cycles, reclosed at 16-1/2 cycles (0.275 seconds) with the fault still present, tripped out again at 21 cycles (0.35 seconds), and locked out. In this case, also, the farm was found to be stable.

Case 3--A 3-phase fault occurred at Location B, near the 69-kV substation terminal of collector line No. 2. Line breaker 4 opened and locked out at 8 cycles (0.133 seconds). All units in the farm were operating fully loaded at 85% power factor prior to the fault. As is shown by the curves in Figure 2, Group 2 units became unstable.

Case 3A--System conditions are those for Case 3; however, a salient-pole generator model was used for each MOD-II unit simulation. As in the previous case, Group 2 units became unstable.

Case 4--With all wind farm units operating at full load and 85% power factor, and without a fault on the system, 69-kV breaker 5 for collector line No. 1 was opened and locked out at 0.0 seconds. In this case, the result was that Group 1 became unstable.

Case 5--A 3-phase fault at Location B cleared normally in 8 cycles (0.133 seconds) as described for Case 3. The prefault unit loadings were varied as follows: Group 2, one-half load at 85% power factor; Subgroup 1A, two-thirds at 85% power factor leading; 1B, zero power and zero reactive; 1C, one-third at 85% power factor lagging; 1D, full load at unity power factor. With differing loading conditions but the same 3-phase fault as in Case 3, the result was also instability among the Group 2 units.

Case 6--A 3-phase fault occurred at the 4,160-volt terminal of a WECS unit step-up transformer. The result was that its 69-kV transformer fuse blew at 30 cycles (0.50 seconds), causing the faulted unit to become unstable. However, the remaining units were stable.

As a result of these simulations, an opening of a collector line circuit breaker for any reason should automatically initiate a shutdown of each unit on that collector line in order to prevent instability among the units connected to that collector line. Further studies have shown that using 12 fully loaded generators lumped at the end of a 1-mile collector line and a 3-phase fault on the other collector line within 1 mile of the 69-kV bus could cause instability among all the units in the farm.

PROTECTION, CONTROL, AND REGULATIONS OF WECS

TVA's standard protection and control of existing 161-kV substations would be used for the WECS collector substations if large wind farms were actually installed on the TVA system. The control of the collector substations would be under the control of their respective ADCC, and the overall power system would be under the control of the PSCC as described earlier. The following control strategies would be adopted for the WECS as a result of the transient stability studies:

(1) Automatic reclosing would not be allowed after the opening of a 69-kV collector line circuit breaker.

(2) All 69-kV breakers at a collector substation would be opened upon a breaker failure at the adjacent 161-kV line.

(3) The opening of a 69-kV breaker would shut down all WECS connected to that 69-kV collector line.

(4) Two separate transfer-trip transmitters would be provided at each collector substation for each farm of WECS, one to shut down units if the 69-kV collector line were opened, and the other to permit the ADCC dispatcher to control each WECS unit.

(5) A transfer-trip transmitter would be provided at each WECS unit location to trip the remote 69-kV breaker at the collector substation. This would, in turn, shut down all WECS connected to the collector line for a failure of the 4.16-kV low side generator breaker.

A master supervisory control and data acquisition (SCADA) terminal would be installed at each of the ADCC's and a remote terminal would be installed at each WECS substation for data acquisition and control. Control functions would include WECS startup, shutdown, 69-kV and 161-kV relay trip and close, and reactive power control. Data would include kW, kVAR, 69-kV bus voltages, 161-kV breaker status, transformer bank loads, alarms, and wind velocities.

EFFECTS OF WECS ON THE TRANSMISSION SYSTEM

Load flow studies of TVA's primary transmission network were made to determine possible areas of inadequacy due to the hypothetical connection of WECS to the transmission system. Both summer and winter peak conditions were studied. The worst case contingency considered was the loss of a 900-MW fossil-fired unit and a 20% loss of the WECS installed capacity during a winter peak. No overloaded lines occurred in any instance, although in some instances additional reactive power control was required.

Estimation of system fault contribution by a single farm is as follows: (1) 342 MVA to the 69-kV collector substation bus and (2) 239 MVA to the 161-kV collector substation bus.

The maximum 3-phase fault contribution from WECS for a 161-kV bus fault with two farms at any given location was found to be 479 MVA. These contributions to system faults would not create any adverse effects on existing equipment for connecting up to 8,520 MW (30% penetration) of WECS generation into the 161-kV transmission system.

V. OPERATION OF THE TVA SYSTEM WITH WIND GENERATION

This section will deal with the preliminary results and the tasks which are currently underway. This phase of the study is concerned with the simulation and the analysis of the operations of the TVA system with wind generation.

SIMULATION OF OPERATIONS WITH WIND

In this simulation it will be assumed that the long-range transmission planning outlined in the previous sections have been completed and the WECS are online at any penetration level between 5% and 30%. At 5% penetration the generating capacity of WECS will be 1,420 MW. Using appropriately scaled wind data from the TVA region, the energy from WECS will be computed by the Wind Performance Model (6). Wind generation will be treated as negative load. The new generation requirements will be used in the unit commitment procedure and the preschedule program. The resulting generation mix, the minimum and maximum energy generated from fossil-fired, nuclear, and hydro units, imports and exports will be compared to those without wind generation. This procedure will be repeated for various penetration levels up to 30%.

ANALYSIS OF THE OPERATIONS OF THE TVA SYSTEM WITH WIND GENERATION

In the unit commitment procedure, the net system load, operating reserves, and minimum operating level of the various units are used in determining the loading levels. The results of the simulation will be analyzed to establish relationships between the operating reserve requirements and WECS penetration levels. The unit dispatch program will treat wind generation as negative load. A sensitivity analysis will attempt to relate changes in the hourly unit dispatch for hydro, fossil-fired, and nuclear units to the changes in the mean wind speeds at various penetration levels.

VI. CONCLUDING REMARKS

The TVA/DOE project to study the impact of significant amounts of wind generation on the operation of an electric power system is still ongoing. The preliminary report of the first two phases of this 3-phase study documents the TVA power system and outlines the reconfiguration requirements for the introduction of wind generation into an electric power system grid. In particular, possible operating procedures for protection and control of WECS collector substations were developed.

In the ongoing phase of the overall study, the daily operation of the electric power system with wind generation is being simulated. The effects of different penetration levels of WECS into the power system on operating reserves, minimum operating levels of various units, unit commitment, and the preschedule sensitivity are being investigated.

VII. ACKNOWLEDGEMENTS

This work is being sponsored by the U.S. Department of Energy (DOE), Office of Electric Energy Systems, through an Interagency Agreement No. DE-A-I01-79-ET23077, entitled "The Impact of Wind Power Generation on the Operations of an Electric Power System." We gratefully acknowledge the technical support of Dr. Jeff Rumbaugh of DOE. Also, the authors wish to thank Leo Edwards, Floyd Watts, Mitchell Needham, and Keith Morris (all of TVA) for their contributions.

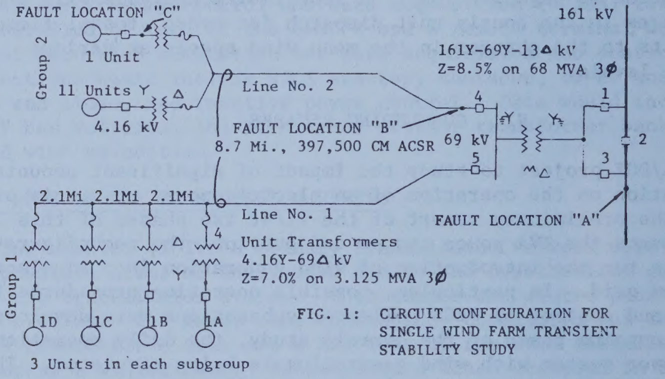
REFERENCES

1. "The Impact of Wind Power Generation on the Operations of an Electric Power System," TVA, Electrical Systems Group, IA No. DE-A-I01-79-ET23077, 1979.
2. B. W. Jones and P. M. Moretti, "Evaluation of Wind Generator Economics in a Load Duration Context," Energy Development III. IEEE Power Engineering Society Papers, 77CH 1215-3-PWR, 1977.
3. Hee Man Bae and M. D. Devine, "Optimization Models for the Economic Design of Wind Power Systems," Solar Energy, Vol. 20, pp 469-481, 1978.
4. "TVA/DOE Analysis of the Operation of an Electric Power System With and Without Wind Generation," TVA, Electrical Systems Group, under preparation.

5. "Large Wind Turbine Design Characteristics and R&D Requirements," NASA Conf. Publ. 2106, DOE Publ. CONF-7904111 Workshop at Lewis Research Center, Cleveland, Ohio, April 1979.
6. Marsh, W. D., "Requirements Assessment of Wind Power Plants in Electric Utility Systems," EPRI ER-978, Vol. 2, January 1979.
7. Krause, P. C. and Man, D. T., "Transient Behavior of a Class of Wind Turbine Generators During Electrical Disturbances," to be published.

DISCLAIMER

"This paper was prepared by the authors. The contents do not necessarily reflect the views and policies of TVA, nor does TVA make any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of any information contained in this paper."



FAULT LOCATION "C"
Group 2
1 Unit
11 Units
Y
Δ
4.16 kV
Line No. 2
FAULT LOCATION "B"
8.7 Mi., 397,500 CM ACSR
161 kV
161Y-69Y-13Δ kV
Z=8.5% on 68 MVA 3Ø
4
1
2
69 kV
5
3
FAULT LOCATION "A"
2.1Mi 2.1Mi 2.1Mi
Line No. 1
Δ
4
Unit Transformers
4.16Y-69Δ kV
Z=7.0% on 3.125 MVA 3Ø
Y
Group 1
1D
1C
1B
1A
3 Units in each subgroup

FIG. 1: CIRCUIT CONFIGURATION FOR SINGLE WIND FARM TRANSIENT STABILITY STUDY

TABLE 1: SUMMARY OF FORECAST DIFFERENCES AND RESPONSES

DAY/HOUR (CST)	DEVIATION	CAUSE	OPERATOR ACTION
4/15/1800 SATURDAY	NSL ESTIMATE: 10,550 MW NSL ACTUAL: 11,254 MW NSL DEVIATION: 704 MW	TEMPERATURE FORECAST DIFFERENCE OF ABOUT 6°F	(i) RAISE HYDRO (ii) IMPORT 693 MW (iii) GENERATE 1,663 WITH COMBUSTION TURBINES (CT)
4/15/2300 SATURDAY	NSL ESTIMATED: 9,620 MW NSL ACTUAL: 10,244 MW	TEMPERATURE FORECAST DIFFERENCE OF 10°F	(i) RAISE HYDRO (ii) IMPORT 361 MW (iii) GENERATE 272 MW with CT
4/15/1400 SATURDAY	THERMAL GENERATION SCHEDULED 10,127 MW ACTUAL THERMAL GENERATION 8,000 MW GENERATION DEVIATION 2,127 MW	LOSS OF A NUCLEAR AND FOSSIL FIRED UNITS	(i) RAISE HYDRO (ii) IMPORT 398 MW (iii) GENERATE WITH CT 480 MW

TABLE 2: MOD-II ELECTRICAL AND MECHANICAL DATA

SYNCHRONOUS GENERATOR: 3125 KVA, 3-PHASE, 60 HZ, 1800 RPM, 4160 VOLTS, 0.80 POWER FACTOR

REACTANCES (PER UNIT): $X_d = 1.931$ $X'_d = 0.252$ $X''_d = 0.144$
$X_q = 1.01$ $X'_q = 1.01$ $X''_q = 0.144$ $X_{al} = 0.081$

FIELD RESISTANCE (PER UNIT): $R_f = 0.00091$ AT 25°C

TIME CONSTANTS: $T''_d = 0.0225$ SECONDS $T''_q = 0.0225$ SECONDS

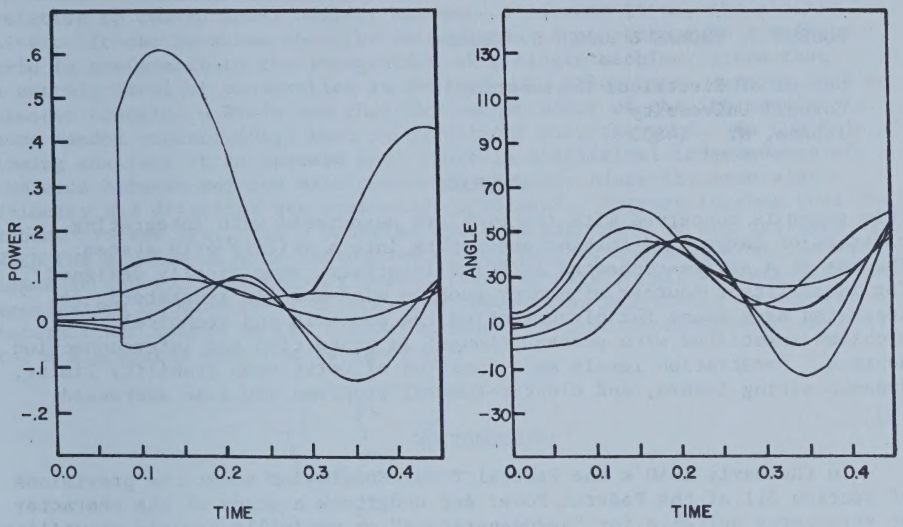
SATURATION FACTORS: $S(1.0) = 0.1765$ $S(1.2) = 0.4712$

ROTATIONAL INERTIA: COMBINED WITH GEARBOX. SEE BELOW.

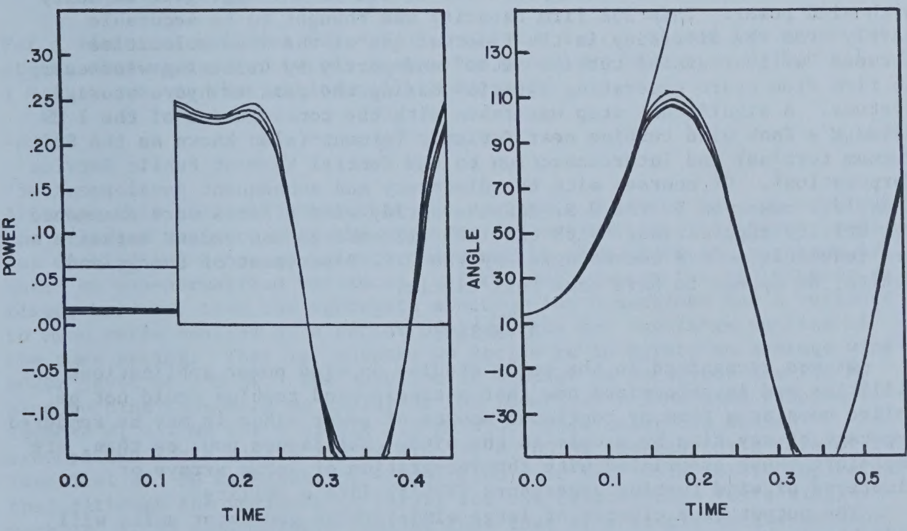
ROTATIONAL INERTIAS (17.55 RPM)

TURBINE BLADES, HUB, AND QUILL: 21.42×10^6 LB -FT -SEC ^{2}/RAD. GEARBOX AND GENERATOR: 1.141×10^6 LB -FT -SEC ^{2}/RAD.

SHAFT STIFFNESS (17.55 RPM): 16.6×10^6 LB -FT /RAD. SHAFT DAMPING (17.55 RPM): 1.0×10^6 LB -FT -SEC/RAD.



CASE STUDY I THREE-PHASE FAULT AT 161-kV LINE TERMINAL. LINE BREAKERS TRIP AT 4-1/2 CYCLES. NO RECLOSURE.



CASE STUDY 2 THREE PHASE FAULT ON 69-kV LINE NO. 2 AT CIRCUIT BREAKER. BREAKER OPENS AT 8 CYCLES. CLASSICAL GEN MODELS. FULL LOAD AT 85 % PF.

Fig. 2 POWER AND SWING CURVES FOR CASES I AND 2 OF THE STABILITY ANALYSIS

INTEGRATION PROBLEMS WITH LARGE WIND ENERGY CONVERSION SYSTEMS

ROBERT J. THOMAS & JAMES S. THORP

School of Electrical Engineering
Cornell University
Ithaca, NY 14853

ABSTRACT

This paper is concerned with the problems associated with integrating clusters of large wind-turbine generators into a utility grid system. The use of a series-connected AC/DC/AC interface, specifically designed for intermittent sources of energy such as wind-turbine generators, is suggested as a means for circumventing the economic and technical problems associated with present thought in protection and interconnection schemes. Penetration levels as a function of worst-case stability limits, dynamic siting issues, and cluster-control problems are also addressed.

INTRODUCTION

In the early 1940's the Federal Power Commission under the provisions of section 311 of the Federal Power Act undertook a study of the character of structures suitable for "aerogenerators" or windmills devoted to utility power supply. This study led to the publication of a survey [1] which concluded that the wind is a stable and reliable source of utility power when developed on a large scale and if operated in cooperation with interconnected steam and hydro systems. At this time the prospect was envisioned of supplying ultimately very cheap power up to perhaps 25% to 30% of a utilities total energy and 15% to 20% of the aggregate capacity with wind power. This 20% firm capacity was thought to be securable partly from the diversity in the fluctuations of the wind velocities between "well separated turbine sites" and partly by utilizing wind energy to firm free hydro generating capacity during the peak in hydro storage systems. A significant step was taken with the construction of the 1 MW Grandpa's Knob wind turbine near Rutland, Vermont (also known as the Smith-Putnam turbine) and interconnection to the Central Vermont Public Service Corporation[1]. Of course, with the discovery and subsequent development of cheap oil reserves in the U.S. and the world, wind efforts were abandoned for utility application. With the 1973 oil embargo and recent emphasis on new renewable source technologies by the U.S. Department of Energy and others, we appear to have come full circle.

DIVERSITY

It was recognized in the early studies on wind power applications for utilities and is recognized now that a single wind turbine could not be relied upon as a firm or continous source of power since it may be rendered impotent at any time by a calm in the wind. The issues now, as then, are therefore those associated with the integration of large arrays or clusters[2] of wind turbine generators (WTG's) into a utility.

The output of a cluster of large wind-turbine generator units will undoubtedly be some aggregate response of specific units and will depend upon the cluster interconnection arrangement, individual unit modal characteristics, the spatial and temporal wind characteristics, the method of cluster control, the penetration level, and the location of the cluster relative to conventional utility hardware placement (i.e., the cluster site). It can be shown that the integration of clusters into a utility grid is preferable to the integration of a single machine, given that a certain level of penetration is desired. To illustrate, suppose that a cluster contains n WTG's and that the output power of the ith machine is some random process $\{Pe_i\}$ that is correlated with the wind. In the following analysis it is assumed that there is statistical independence of the wind between any two machines about the mean since the mean wind velocity and direction are spatially correlated. Suppose further that the random process $\{Pe_i\}$ has a Gaussian spectrum with mean p_i and variance σ_i. With the premise of wind independence, and further assuming that the variance in the spectral density scales linearly with power, we note that the aggregation of many small intermittent sources is preferable to one large intermittent source in the following sense: the variance-to-mean ratio for n aggregated machines is given by

---

[1] Grandpa's Knob Data - Electrical World, October 19, 1940.

[2] By a cluster we mean on the order of fifty or more WTG's.

$$r_T = \frac{\sqrt{\sum_{i=1}^{n} \sigma_i^2}}{\sum_{i=1}^{n} p_i} \tag{1}$$

It follows that the variance-to-mean ratio for one machine is

$$r_T = \frac{\sigma_1}{p_1} \tag{2}$$

For n identical machines ($\sigma_i = \sigma_T$ for all i = 1,....,n) and for operation at the same mean power output (mean wind speed) we have $p_i = p_T$ for all i = 1,....,n, so that equation (1) reduces to

$$r_T = \frac{\sigma_T}{\sqrt{n}\, p_T} \tag{3}$$

Although this analysis shows that the aggregate spectrum for two machines has a larger variance than the spectrum for one machine, the power level has doubled and hence an unfair comparison is made since the spectrum is based on non-normalized variances. If the mean rating for the cluster is fixed, however, then the aggregate spectrum for n machines has a variance to mean ratio smaller by a factor of $1/\sqrt{n}$ than for one large machine of the same rating. That is, suppose we decide to integrate an average wind-power level of 250 MW. The question is whether to integrate one 250 MW wind turbine (assuming we could) or 100 identical 2.5 MW wind turbines. Suppose σ_1 = 1 MW for a single 2.5 MW WTG. Then under the stated assumptions for a single 250 MW WTG, σ_T = 100 MW whereas for a farm of 100 identical 2.5 MW machines, $\sigma_T = 100/\sqrt{100}$ = 10 MW. This result indicates that although the variance is larger for the cluster than for a single machine, it is considerably smaller than that of a single larger machine. Recalling that 2.5σ represents roughly a 99% confidence level, we may be confident that a farm that is operating at 250 MW will continue to operate at 250 ± 25 MW for a "long time" while a single large 250 MW machine could swing 250 MW ± 250 MW. Clearly, cluster steady-state behavior is statistically superior.

PROTECTION

The previous analysis provides a basis for deciding that a cluster of WTG's at a fixed level of penetration (measured as a percent of installed conventional capacity) into a power grid is desirable. The assimilation of these clusters into standard power networks poses important operational and planning problems for power-system engineers because WTG operating characteristics are uncertain due to the intermittent nature of the wind, and because WTG's appear to have no counterpart in established hardware. One of the problems that need to be addressed is the protection of both the WTG cluster and conventional utility hardware.

Present approaches to the integration of clusters seem to be centered on: a T-bus arrangement; a ring-bus arrangement; or sectionalized versions of these two standard interfacing configurations, with security based on standard bus protection. These techniques are reasonable for a few (say five or six) machines, but become economically prohibitive if clusters are contemplated. To illustrate the problem, consider n generators in a T-bus arrangement as shown in Figure 1. Standard bus protection requires circuit breaker (CB_i) ratings to be capable of interrupting the largest fault current (about 2.6[3] times the maximum short-circuit machine current) that could flow. In the case of Figure 1, this condition would exist whenever an internal machine fault occured as shown in the network diagram of Figure 2 for machine i.

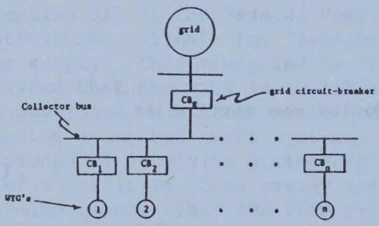


Figure 1 Suggested T-Bus Cluster Arrangement

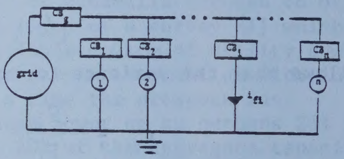


Figure 2 Equivalent Network for T-Bus Cluster

The fault current i_{fi} contains the sum of all the fault-current components of the (n-1) remaining machines plus the grid current. Even if we assume a very large grid impedance (due to, say, a significant transmission-line distance plus step-up transformer), thereby neglecting the contribution of the grid current, i_{fi} can be appreciable. For example, if the collector bus is at generator voltage (4.8 kV for a MOD-2)[4] and there are 101 machines, the short-circuit current contribution, neglecting the grid contribution, is about 300 amperes per machine (i.e., $2.5 \times 10^6/\sqrt{3} \times 4.8 \times 10^3$) or about 30,000 amperes per breaker (300×10^2). Hence the fault current is likely to be 78,000 amperes ($2.6 \times 30{,}000$), which is beyond the capability of state-of-the-art breaker technology. The point is that under this arrangement we must protect each 2.5 MW WTG as though it were a 250 MW or greater generating plant. One alternative would be to raise the terminal voltage of each generator through the addition of a step-up transformer and protect on the high side as shown in Figure 3. However, the transformers are left unprotected from internal generator and/or transformer faults. This is standard utility practice for large generating units because there are no alternatives. The transformers of Figure 3 would be protected as a matter of course if the breaker technology were available. Also, note that for this configuration, the cost of protection must include transformer plus breaker/relay costs. In addition, every breaker would need to open in response to a fault on the collector bus thereby reducing the availability of the cluster significantly.

---

[3] A nominal figure used by circuit-breaker designers.

[4] A MOD-2 wind turbine is a Boeing horizontal axis machine rated at2.5MW.

Some of these problems can be circumvented by sectionalizing the WTG system based upon generator size and circuit-breaker interruption capability. For example, the system of Figure 4

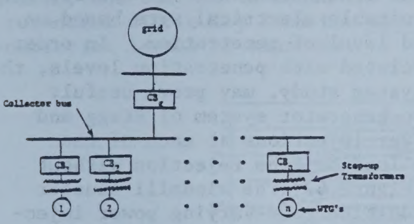


Figure 3 T-Bus Cluster With Step-up Transformers

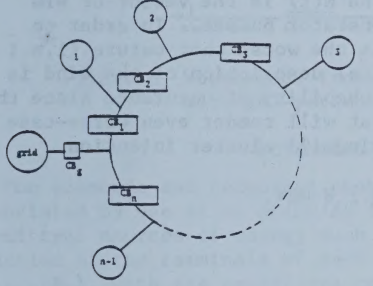


Figure 5 Ring-Bus Cluster Arrangement

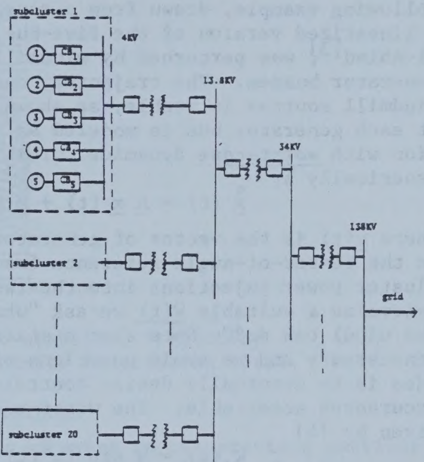


Figure 4 Sectionalized T-Bus arrangement showing Sub-clusters and Step-up Transformers

would be acceptable from a protection point of view. This sectionalized system, consisting of 100 WTG's contained in 20 subclusters of 5 WTG's per subcluster, requires 172 circuit-breakers plus 35 transformers of various ratings. This configuration, however, is preferred to the straight T-bus because of the adequate transformer protection and because all circuit-breaker ratings are within present technology capabilities (due mainly to the introduction of fairly substantial transformer impedances between non-adjacent subcluster busses). Preliminary estimates of protection costs(2) for the system of Figure 4 indicate an amount in 1976 dollars (excluding wire) or about \$165/kW or a cost of about \$412,500 per MOD-2 machine. With target fabrication costs of roughly \$1.6M in 1976 dollars per WTG for the MOD-2, excluding installation, the protection and integration costs become a significant part of the overall cost.

Another bus arrangement that has been suggested for large clusters of wind machines is the ring-bus configuration of Figure 5. The advantage of this arrangement is that a machine bus fault can be isolated by opening two circuit breakers and hence the operating reliability of the cluster is enhanced. Each breaker must still be capable of interrupting the sum of the maximum fault currents from each WTG plus the current supplied by the grid, as was the case for the T-bus arrangement. As before, state-of-the-art circuit-breaker capability would be exceeded. Again, the ring bus could be sectionalized based on maximum circuit-breaker ratings, but the cost of protection would be equivalent to that of the T-bus.

DYNAMICS

From a dynamic point of view, the amount of wind power that can be injected into a utility system for a specific site location is limited mainly by stability and quality of service considerations. Of course, the dual problem is the determination of a suitable electrical site based on stability considerations, given a desired level of penetration. In order to demonstrate the dynamical issues associated with penetration levels, the following example, drawn from a recent system study, may prove useful: A linearized version of the five-bus, two-generator system of Stagg and El-Abiad[3], was perturbed by windmill power injections at each of the generator busses. The trajectory due to load or line rejection without windmill sources is stable, as shown in Figure 6. The windmill cluster at each generator bus is modeled as an additive time-varying power injection with worst-case dynamics. That is, the system description is given generically by

$$\overset{o}{\underline{x}}\,(t) = \underline{A}\;\underline{x}\,(t) + \underline{B}\;\underline{W}(t)$$

where $\underline{x}(t)$ is the vector of generator torque angles and speed deviations in the center-of-angle reference frame and $\underline{W(t)}$ is the vector of WTG cluster power injections into the two generator busses. In order to determine a suitable $\underline{W(t)}$ we ask "what is the worst that nature (i.e., the wind) can do?" Note that a statistical description of the wind is unnecessary and we avoid questions of probability of occurance since the idea is to eventually design controls that will render even worse-case occurances acceptable. The worst-case windmill cluster injection given by [4]

$$W_i(t) = \tfrac{1}{2}\,\text{sgn}\,(\underline{x}^T(t)\;\underline{P}\;\underline{B}\,)\;W\,\max_i$$

$$\text{where sgn }(z) = \begin{cases} +1 & z > 0 \\ \;\;0 & z = 0 \\ -1 & z < 0 \end{cases}$$

and $\underline{P}$ is the solution to Lyapunov's equation

$$\underline{P}\;\underline{A} + \underline{A}^T\underline{P} = -I$$

Figure 7 shows the trajectory at generator bus 1 for a penetration level of 1.25 MW (1.47% of total generation) at each bus.

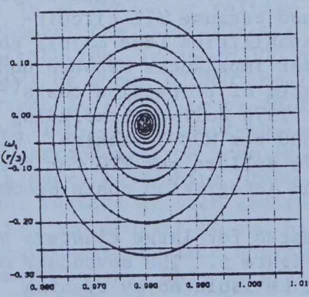


Figure 6 Typical Tranient-stability Trajectory for the Stagg and El-Abiad System Without Windmills

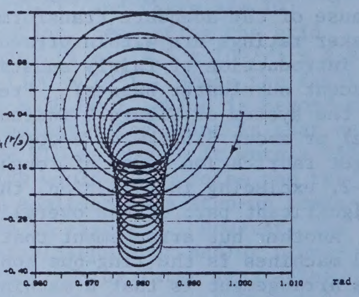


Figure 7 Worst-case Transient-stability Trajectory with 1.47-percent Windmill Penetration Level.

The disconcerting presence of undamped oscillations is clearly indicated. Figure 8 shows the trajectory for the relatively uncharacteristic case of 125 MW (147% of total generation) at each bus. The undamped oscillations are further emphasized. While the model for these two preliminary cases may not be entirely valid because it is linearized, additive, and not rate limited, the concept is sound and the issue of dynamic stability represents a potential difficulty from the system point of view and with respect to penetration levels of clusters.

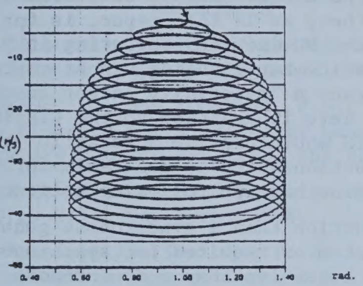


Figure 8 Worst-case Transient-stability Trajectory with 147-percent Windmill Penetration Level.

POTENTIAL SOLUTIONS

The economic and technical problems adressed in the previous sections are obviated by use of an AD/DC/AC interface specifically designed for intermittent sources of energy such as WTG's. We have in mind the connection of the terminals of each machine to an uncontrolled rectifier $(R_1, \ldots, R_n)$, with the rectifiers connected in series. The resultant output would be connected to a controllable inverter system (CIS) coupled to the grid as shown in Figure 9. The CIS must be designed to interface with the grid at constant voltage. The input voltage to the CIS would be variable, depending upon the outputs of the several WTG's, and could range from zero (no wind or excessive wind at or beyond cutout velocity at every WTG site) to 480kV (assuming 100 MOD-2 machines operating at rated wind at every WTG site). Thus the grid voltage would be constant while the inverter output voltage would be adjustable in response to the change in input voltage as the system absorbs whatever power the cluster has available. This arrangement has the following inherent advantages:

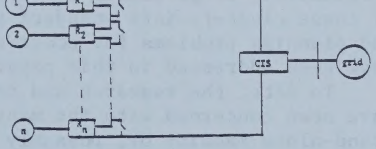


Figure 9 Suggested Cluster Coupled to the Grid by a Controllable Inverter System (CIS).

1. Generator interactions are eliminated;
2. Resynchronization is unnecessary;
3. Individual generator speeds may vary widely, which means that the cluster would be able to capture significantly more wind because of the ability of each machine to run asynchronously;
4. The system is self-protecting;
5. The output of the cluster can be centrally controlled without communication devices for each machine;

6. The system is economically attractive since preliminary cost projections (excluding wire and grid interconnection equipment) are about \$5/kW for the rectifiers and \$50/kW for the inverter, or about one-third conventional a-c system protection costs.

Point 3 in the list of advantages deserves some elaboration. Virtually all recent designs of wind turbines are constant speed synchronous machines sized differently for different average velocity wind regimes. In the early 1940 Federal Power Commission study [1] it was recognized that an economical set of WTG's naturally operates at a speed dependent on the velocity of the wind while the energy must be delivered at constant frequency. The suggestion then, as in this paper, is for DC/AC conversion (although the survey suggested DC machines operating in parallel with individual inverters, automatic tap-changing transformers, and field regulation) in order to extract a "materially larger amount of total energy from the wind". We suggest here in addition to the CIS interface, an induction machine on each WTG would provide a versitile and robust system of inherently simple construction which is desirable for WTG cluster systems. The induction machine has the following salient features:

1. It is simpler in construction than a synchronous generator.
2. It has no excitaiton system as required for synchronous machines.
3. It is considerably less expensive than a synchronous generator.

While there are a number of non-trivial problems associated with this system, success in the design would significantly impact present design philosophy of large wind turbines.

CONCLUSION

In view of recent research and development successes in large wind-energy conversion generation technology, it seems inevitable that this new energy source technology will be integrated into existing power-system grids during this decade. Of the units intended for commercialization at present, the Boeing MOD-2, a horizontal axis, 2.5MW machine appears to be the most highly developed. This design appears to be at the limit of power output for a single machine because of aeroelastic stability problems that would occur at higher megawatt levels. Thus, in order to inject substantial amounts of power successfully into a utility system, a relatively large number of machines will be required. The assimilation of these clusters into standard power networks poses important operational and planning problems for power-system engineers. Some of these problems have been addressed in this paper.

To date, the research and development efforts applied to large WTG's have been concerned with the many specific problems associated with a stand-alone machine or, possibly groups of two and three machines. That is to say, design issues have not included those of cluster integration (such as stability, protection, and control) except in the case of very low penetration levels (a few machines against an infinite bus). These investigations have generally been confined to the areas of efficiency, cost, and machine-mode identification and interaction. Only cursory attention has been paid to the compatibility between an existing power network and a cluster supplying a substantial portion of the system load. Consequently, little is known regarding the expected operation of wind-energy conversion systems (WECS).

In order to assess the effects of large WECS relative to system considerations, issues of siting as a function of penetration levels, aggregate dynamic models, and protection will need to be addressed from the composite power-system point of view as opposed to the single WTG point of view where the system is represented as an infinite bus. Any new concepts derived from this broader viewpoint would be useful in the design of future generations of wind machines.

ACKNOWLEDGEMENT

We would like to thank Mr. John W. McClure, a graduate student in the School of Electrical Engineering for his valuable assistance in generating Figures 6-8.

REFERENCES

1. Thomas, P.H., "Electric Power from the Wind-A Survey", Federal Power Commission, Office of the Chief Engineer, March 1945.
2. Linke, S., Teshome, A., and Yehsakul, P.D., "A Study of Transmission and Protection Elements for Wind-Energy Generating Systems", Brookhaven National Laboratory reports BNL 50851 and BNL 24345, April, 1978.
3. Stagg, G.W. and El-Abiad, A.H., "Computer Methods in Power System Analysis", McGraw-Hill, New York, 1968.
4. McClure, J.W. "Destabilization of a Power System with Bounded Uncertainty", A Masters Thesis, Cornell University, August 1980.

POWER SYSTEM PLANNING AND THE NEW TECHNOLOGIES

R. L. SULLIVAN
Department of Electrical Engineering
University of Florida, Gainesville, FL 32611

Abstract: This paper discusses some of the problems associated with modeling and simulating the impact of integrating selected new intermittent technologies into modern electric energy systems. In particular, the difficulties that arise in forecasting future load requirements, in planning new capacity additions, and determining production costs for expansion plans containing new technologies are presented. In most instances the discussion is couched in the framework of current power system planning practices in order to identify near term modeling and simulation issues. The paper is by necessity both quantitative and qualitative, and concludes that current planning techniques can conceptually only handle centralized new technologies. Much more effort is needed to provide planning tools that can properly evaluate the new intermittant technologies that are distributed throughout the electric energy system or that have strong chronological characteristics.

INTRODUCTION

The paper begins in this section with an overview of power system planning including a discussion of the various simulation methodologies in use and under development.[1] This overview is followed by the discussion of (1) the centralized new technologies, (2) the dispersed technologies, and finally (3) the conclusions. It is hoped that this organizational structure will assist the reader in understanding the problems that must be addressed in order to evaluate the impact that the new intermittent technologies as well as load management will have when integrated into existing systems.

Power system planning has historically been decomposed into three subtasks: (1) Generation System Planning, (2) Transmission System Planning, and finally, (3) Distribution System Planning. Generation system planning basically involves determining the future load requirements and the most economical mix of generating units to meet these requirements reliably. The tools used in generation system planning have evolved from deterministic analyses to probabilistic and stochastic analyses In the fifties and early sixties, capacity additions were determined using outage capacity tables that considered the forced outage characteristics of the candidate units, although the load was usually represented deterministically. The results of this activity provided the alternative generation expansion plans to be subjected to deterministic production analysis to determine the production costs and general operating impacts of each alternative plan. The production analysis was, in effect, a real-time simulation of the hourly operation of the system over the planning horizon. As such these techniques did not represent the forced outage characteristics of the generating units, did not insure consistency between the demand and energy requirements, did not provide measures of reliability, and were computationally expensive (relative to the cost of computation at the time). Needless to say, the results obtained from these early techniques were inaccurate, and parametric analysis expensive - thus in the mid to late sixties, probabilistic production cost analysis replaced the ineffective deterministic techniques. Probabilistic production analysis techniques represent a major departure from the earlier real-time simulation methods. To improve accuracy, the forced outage characteristics of the units are explicitly considered as are their maintenance schedules etc.,. To overcome the computational barriers, a more analytical structure was adopted that enabled the planner to simulate intervals of time from a week to a year instead of hour by hour - and to do so accurately. Furthermore, these techniques facilitated insuring consistency between demand and energy forecasts and enabled planners to model forecast uncertainty. Although this approach to generation system planning has been quite successful, the introduction of the intermittent new technologies and load management has created computational problems which will be discussed throughout the paper.

Transmission system planners, as well as distribution planners, have relied very heavily on deterministic tools. In the fifties, T & D expansion plans were partially assessed using analog network analyzers. The introduction of high speed digital computers replaced the network analyzer, and now planners have power-flow, transient stability, short-circuit, and transient overvoltage computer programs to use in planning system expansions. None of these tools address the reliability issue in a probabilistic manner, i.e., there exist few tools at this time that have had the same impact on T & D planning as the probabilistic production analysis tools have had on generation system planning. Reliability analysis, in particular, and T & D capacity planning, in general, suffer from dimensionality problems since it takes 2N (N = # of busses) nonlinear equations to describe the performance of the power network. The lack of a probabilistic technique for T & D planning is the main reason why generation and distribution system reliability analyses have evolved along two different lines. Furthermore, without techniques for translating network level decisions in terms of equivalent generation system impacts, it will be difficult to properly analyze the cost-effectiveness of displacing generation with dispersed technologies. Although work is underway to develop probabilistic power-flow and transient stability analysis techniques which consider the randomness of system loads and the outage characteristics of the many network elements, the theoretical problems remaining are significant. As will be discussed later, the development of the array processor may resurrect stochastic methods for dealing with T & D planning.

As alluded to in the above paragraphs, system planning has evolved from the use of primarily deterministic tools to the use of a combination of deterministic and probabilistic methods. Of the three planning subtasks, the T & D planning subtasks have benefitted very little from the shift away from deterministic techniques. Because of the difference in level of analysis capability between generation system planning and T & D planning, generation systems are still planned without proper consideration of the effect of the network elements and T & D systems are planned without considering, in a mathematically consistent manner, the tradeoffs to be realized between generation capacity and performance and T & D capacity and performance. In short, there exists no bulk power system reliability and cost analysis technology that enables the utility industry to identify the optimum expansion plan, the plan for which the marginal costs equals the marginal benefits.

PLANNING FOR WECS AND PVS

In order to assess the applicability of current planning methods to evaluate the impact of integrating the new technologies into existing systems, it is necessary to first discuss the salient characteristics of the technologies of interest. In particular, this paper will address both centralized and decentralized energy conversion systems (WECS) and photovoltaic systems (PVS). Although there are other new technologies - Ocean Thermal Energy Conversion (OTEC), Fuel Cells (FC), and even Load Management (LM) - WECS and PVS more than any of the others except for LM stress the capability of current planning tools. Since LM is a very near term technology, a few comments will be made when convenient concerning the simulation problems arising when attempts are made to assess its impact.

A. WECS and PVS Model

Since the primary resources driving both WECS and PVS are highly random, the available capacity from these technologies is also random. Due to the nonlinear relationship between resource and power output the use of analytical methods to derive probability density functions for available capacity is not straightforward, thus many different output models are used - from a deterministic hourly average capacity to a psuedo Monte Carlo procedure where the output capacity is weighted by the availability of the corresponding resource, and then averaged.[2,3] From a more rigid mathematical view, it is desireable to determine a model for WECS and PVS that represents the availability of the resource and the outage characteristics of the systems, i.e., it is desirable to have the available capacity density function $f(C_s)$, where C_s is the available capacity to serve load. To obtain $f(C_s)$, let

$$C_s = C - C_o \tag{1}$$

where C is the capacity available from the basic resource, and C_o is the outage capacity due to the failure of the system. f(C) can be determined by analyzing resource data and relating the data to the available electric capacity. Similarly, $f(C_o)$, in its simplist form, can be constructed knowing the forced outage rate "q", and availability "p". Recognizing that C_o and C are directly correlated with a coiefficient of correlation equal to unity,

$$f(C_s) = \sum f(C_o/C)\ f(C) \tag{2}$$

For discussion purposes, if it is assumed that C can take-on only three values C_1, C_2, and C_3 then $f(C_s)$ is defined as shown below:

C_s	$f(C_s)$
0	q
C_1	$pf(C_1)$
C_2	$pf(C_2)$
C_3	$pf(C_3)$

Table 1.

This result suggests that the proper model for WECS and PVS considering their failure characteristics is determined by the derated, if you will, values describing the probability that they are operating in a given capacity state, one of which could be zero (which would simulate the unavailability of solar insolation at night for example). Further, there is a fourth state representing the failure of the WECS and PVS. Knowing $f(C_s)$, it is straight forward to obtain the more common outage load model used in current reliability/production costing codes, i.e.,

l_o	$f(l_o)$
0	$p/3$
C_1	$q/3+ p(1-f(C_1))/3$
C_2	$q/3+ p(1-f(C_2))/3$
C_3	$q/3+ p(1-f(C_3))/3$

Table 2.

Depending on how one incorporates the WECS and PVS into the capacity planning problem, either $f(C_s)$ or $f(l_o)$ would be used, a point that will be discussed in some depth in Section B.

B. CENTRALIZED WECS AND PVS

Assuming that it is of interest to assess the impact of central station WECS/PVS systems, then the planner is faced with a supply-side evaluation. To model these new technologies as central station technologies, standard concepts can be employed.[4] Let L represent the generic load and f(L) its probability density function, and, let L and $f(L_o)$ represent the outage load and associated density function for the conventional units. The distribution for the effective load would be:

$$F(L_E) = \sum F(L_E - L_o)\ f(L_o) \qquad (3)$$

Convolving-in the WECS and PVS outage load density function, $f(l_o)$, the final effective load distribution against which all units would be committed becomes:

$$F(L_E) = \sum\sum F(L_E - L_o) f(L_o) f(l_o) \qquad (4)$$

where $f(l_o)$ is given in Table 2 for the sample three state WECS/PVS. An equivalent way to write eq. 4 assuming that the new technologies would be committed first is:

$$F(L_E) = \sum\sum F(L_E - L_o - l_o) f(l_o) f(L_o) \tag{5}$$

if the load and l_o are assumed independent or

$$F(L_E) = \sum\sum F(L_E - L_o - l_o/l_o) f(l_o) f(L_o) \tag{6}$$

if L and l_o are assumed to be correlated.

Another approach, which is frequently used, is to subtract-off the WECS or PVS capacity from the generic load before convolving-in the conventional units to determine reliability and production costs, i.e., define the residual load L_R as:

$$L_R = L - C_s \tag{7}$$

then,

$$f(L_R) = \sum f(L/C_s)\ f\ (C_s) \tag{8}$$

$f(L/C_s)$ is the conditional probability density function that captures the dependence between the generic load and the WECS or PVS capacity. To determine the density function, the following identity may be useful:

$$f(L/C_s) = \frac{f(L,C_s)}{f(C_s)} \tag{9}$$

$f(L,C_s)$, the joint density function for L and C_s, could be obtained by analyzing load and coincident resource data. $f(C_s)$ is given by eq. 2. Knowing $f(L_R)$, the effective load to be served by the remaining conventional units would be defined as:

$$L_E = L_R + L_o \tag{10}$$

and

$$F(L_E) = \sum F(L_R)\ f(L_o) \tag{11}$$

$F(L_R)$ is the load duration curve for the residual load and can be obtained by first determining the residual chronological load curve and accumulating the hourly loads to obtain the load duration curve - a procedure that is used in most reliability/production costing codes. Upon expanding eq. 11, it can be written as:

$$F(L_E) = \sum\sum F(L_E + C_s - L_o/C_s) f(C_s) f(L_o) \tag{12}$$

and compared to eq. 6. Equation 6 describes the probability that the effective load L_E will exceed any load level X, and eq. 12 is the probability that the effective load will exceed a residual load level y. Since the LOLP (loss of load probability) is calculated by determining the probability that the effective load exceeds the installed capacity of the system, eq. 6 must be evaluated at $L_E = IC + C_s$ but eq. 12 must be evaluated at $L_E = IC$, where IC is the installed capacity of the conventional system. Furthermore, these two formulations only give the same results if $F(L_R)$ equals the effective load duration curve after the WECS or PVS have been committed to serve load. In either case, it is probably important to recognize the dependency between C_S and L, and to capture this dependence through the conditional density functions defined in eq. 6 and 12. Finally, for purposes of determining production costs, considerable error could be obtained if the dependency between L and C_S is ignored, since it is through the conditional density functions that the time dependence or time correlation between L and C_S is captured.

C. DISPERSED WECS AND PVS

If dispersed WECS and PVS become a reality and they are viewed as consumer-owned demand-side technologies, then, load forecasters will by necessity be required to take into account their effects on the residual load to be served by the conventional units. Similarly, generation planners will be required to consider the need for back-up capacity in determining the optimal mix to meet the future load requirements. T & D planners will face the problems associated with integrating the dispersed technologies into the existing system, problems which will be aggravated by the highly random nature of WECS and PVS capacity. Finally, corporate analysis of WECS and PVS will require production results to determine their financial impact and, if necessary, to consider modifications to existing rate structures to recover fixed costs associated with providing back-up capacity.

Forecasting future load requirements when dispersed technologies become significant, will greatly increase the complexity of forecasting. In general it can be anticipated that the uncertainty associated with load forecasts will increase in view of the intermittent nature of the candidate dispersed technologies. Since the availability of the new technologies, and hence, the capacity of the new technologies will be correlated with the generic system load, it is quite likely that forecasters will attempt to forecast L and C_S separately, and then determine $f(L_R)$, the residual load probability density function, using ideas presented earlier. Furthermore, in view of recent legislation, each consumer may also become a cogenerator, thus further complicating the determination of future load requirements. Attempting to develop new forecasting methods at this time to deal with dispersed technologies seems premature, and especially in view of the current state-of-the-art. To properly reflect the impact of dispersed technologies, reliability and production analyses should include the effects of the T & D system by considering the entire bulk system. Although bulk system reliability analysis is undeveloped relative to generation system analysis, there are some promising new ideas being considered. One such idea uses two-state component models coupled with a least square error approximation to the AC power flow model to formulate the problem as follows:[5]

$$\min \sum_{y \in Y} Y_i$$

$$\text{St. } AX \geq P_T + Fy \tag{13}$$

where, Y_i = i^{th} topological state of the system

P_T = system bus load vector

F = capacity matrix

X = system voltage vector

Solving the above problem identifies those paths through the network that have the least number of elements but yet still satisfy the linearized power-flow equations for a given load level P_T. By increasing the load level until the power-flow equations are violated each minimum path set can be associated with the maximum load it can serve - such sets contain so-called load dependent minimal paths. Through repeated applications of the above, the entire state space can be decomposed into sets of minimum paths and their associated maximum load carrying capabilities. Since the probability that the system will reside in a given minimum path is a single multiplicative combination of the individual component availabilities and forced outage rates, it is possible to determine the probability that the bulk system is capable of supplying a load greater than or equal to the installed capacity - which is LOLP. Further, the expected value of demand not served can also be determined as an additional measure of system reliability. Since this technique includes the T & D system, it would more accurately represent the impact that dispersed technologies would have on system reliability. Further, the technique has the capability to identify the source of system failures, thus planners can determine if a failure is due to generation or transmission limitations - an important feature when evaluating new technologies. If the failure is due to a transmission system limitation, the technique can determine if it is a voltage problem or a transfer limit problem. This simulation capability is extremely important when trying to assess dispersed technologies which can operate in a cogeneration mode.

Reference [6] describes another bulk power system reliability analysis tool which uses a fast decoupled power-flow program to compile the failure effects of single-contingencies for various random load levels. Using these results plus a procedure for determining the probability that the system will reside in a given load/system state, a variety of reliability measures can be determined. The approach taken here is less analytical than the technique described in reference [5] as it approaches Monte Carlo simulation in a fashion.

LOAD MANAGEMENT

Load management can be decomposed into two subcatagories: Direct and Indirect. Load management through direct load control (DLC) is very common in Europe, and is being considered very seriously by utilities in this country.[7] Since the main purpose of DLC is to modify the system load shape to provide the same level of reliability at lower costs, it naturally requires controlling loads to (1) avoid , in the near term, using high cost peaking capacity by load shedding non-critical loads and, (2) avoid, in the long term, the need for large amounts of peaking capacity by reducing system peak requirements through load control. To evaluate the effect of a given DLC regime both reliability and production cost analysis tools must be capable of tracking the system state on an hourly basis to effect the necessary control. For example, if the DLC strategy being considered involved shedding load when certain peaking units are required to start, then the technique used would have to first execute a commitment algorithm, and if the peaker in question is committed, to back-up and shed load instead. In other words, current reliability and production simulation tools that can only preprocess the chronological load data to take into account load modifying technologies, such as DLC, are simply inadequate. Similarly, time of day (TOD) pricing schemes being discussed to achieve an optimal allocation of resources, also modify the system load. Since the most talked about TOD regime is peak-load pricing, hourly modifications of system load are again required.

CONCLUSIONS

In view of the intermittent nature of WECS and PVS, it is reasonably clear that to evaluate their impact, bulk power system reliability and production costing capability will be required. Current techniques are being used with success in considering central station placements where the T & D system can be ignored as in the past. In most cases the residual load approach is used in order to capture the time correlation effects between the generic system load and the availability of WECS and PVS capacity. It was argued that this approach is identical to the conventional technique if the residual load distribution is identical to the effective load distribution after WECS or PVS have been committed to serve load. The distributions are identical in that one distribution can be obtained from the other through a simple shifting operation, where the magnitude of the shift is equal to the capacity of the new technology.

Finally, in the last section the paper suggests that to simulate the load management technologies that Monte Carlo techniques offer some advantages. If and when array processors become fully developed, the computational expense of using Monte Carlo based methods may disappear, thus enabling planners to more accurately capture both the stochastic features of the new intermittent technologies as well as the chronological characteristics of load management regimes.

ACKNOWLEDGMENT

This work was supported by the Solar Energy Research Institute, under Contract No. CA-0041-1.

REFERENCES

1. Electric Power Problems: The Mathematical Challenge, Proceedings of International Conference Sponsored by SIAM, March 18-20, 1980, Seattle, Washington.

2. Fegan, G.R. and Percival G. D."Problems In the Integration of Intermittent Sources Into Utility Production Costing Models" Proceedings of International Symposium in Circuits and Systems, April 29, 1980, Houston, TX.

3. Fegan, G. R. and Percival G. D. "Integration of Intermittent Sources Into Bateriaux-Booth Production Cost Models," IEEE Winter Power Meetings, IEEE paper #A80 111-5, February 1980, New York, NY.

4. Sullivan, R.L. "Power System Planning" Book, McGraw-Hill, 1977

5. Ranade, S. J. and Sullivan, R. L. "A Reliability Analysis Technique for Bulk System Planning," Paper submitted for 1981 IEEE Winter Power Meeting, Atlanta, GA.

6. Marks, G. M., Director of System Planning, Florida Power Corporation, Personal Communications, 1979-80.

7. Sparkman, R. E. and Sullivan, R. L. "Combined Generation System and Load Management Production Analysis," Paper submitted for 1981 IEEE Winter Power Meeting, Atlanta, GA.

DYNAMIC DATA STRUCTURES : FINITE FILES, LIMITING PROFILES AND VARIANCE ANALYSIS

L. CHENO[1], P. FLAJOLET[2], J. FRANÇON[3], C. PUECH[4], J. VUILLEMIN[5]
[1]Ecole Normale Supérieure, [2]INRIA, [3]Université de Strasbourg,
[4]Université René Descartes, [5]Université Paris-Sud.

Abstract : Dynamic data structures for priority queues, dictionaries... are analyzed under sequences of arbitrary insert, delete and query operations in various contexts : when the universe of keys is finite both exact and asymptotic estimates are provided ; when the universe of keys is infinite, a new asymptotic formula is given which makes it possible to analyze rather complex data structures including binomial queues. We conclude by further showing how to derive variance estimates for simple structures.

INTRODUCTION

Dynamic data structures (dds) can be subjected to various operations resulting in a possible increase or decrease in size. The oldest such structures are certainly linked lists considered in the mid fifties. Lists have the advantage of requiring hardly more storage (one extra pointer per key) than the quantity of information they are holding at each stage. Trees have the further advantage of allowing a more efficient logarithmic search. The importance and use of these structures for performing dictionary, priority queue, linear list of partition ("union-find") operations are too well-known to be recalled here and are described for instance in [5]. Surprisingly enough, analysis techniques -that constitute the object of the present paper- have taken a much longer time to develop and have received comparatively little attention. The reason can probably be assigned to the difficulty of finding "natural" statistics, and, what is more compelling, to the difficulty of carrying out the analyses themselves due to the inherent complexity of the underlying enumeration problems (it is usually a non trivial task to determine the number of configurations corresponding to a given cost). The common pattern of all the existing dynamic analyses is to consider for each integer n the set of all possible input sequences of length n, or a sample subset taken as representative if the former is infinite. Let Ω_n be the finite set of sequences of length n on which the analysis is performed. For a given data structure S together with its companion algorithms and to each input sequence sometimes also called request sequence of sequence of operations s is associated a certain cost usually measured by the amount of time or storage resource required. Denoting this cost by $\mathrm{cost}^S(s)$, the average cost of a sequence of n inputs (or requests) is naturally defined as :

$$\overline{K}_n^S = \frac{1}{\mathrm{card}\,\Omega_n} \sum_{s\in\Omega_n} \mathrm{cost}^S(s).$$

This quantity will be called the integrated cost of data structure S under sequences of n requests (of type Ω).

The present paper is in the line of previous works by the authors [3] to which the reader is referred for some definitions and a more complete bibliography. We are interested in the following questions not considered in other works :

1) Analyze dictionaries, priority queues... under the set of all possible input sequences when the "universe" of possible keys is finite. This models situations where all transactions are relative to a fixed reference file. The case of priority queues appears for instance in an operating system with a predetermined set of users each enjoying a certain priority.

2) Analyze more complex data structures than have been previously considered : the difficulty then is the rather heavy combinatorial expressions for integrated costs which may soon prove intractable. We prove here the existence of limit distributions for the profiles of histories (sampled input sequences). The approach makes it possible to rederive in a simple way asymptotic expressions for all known integrated costs of priority queues. We demonstrate its usefulness by providing a complete study of binomial queues. The analysis reveals rather deep periodicity phenomena in the structure under the effect of arbitrary sequences of inputs.
3) Study the dispersion of costs of dds under sequences of operations. We explicit the case of memory occupation for most structures as well as the number of comparisons for sorted and unsorted list implementations of priority queues and dictionaries.

About the methods employed , the keystone is the use of continued fractions and orthogonal polynomials whose first appearance in the context of dynamic structures goes to [2], [3]. Profile analysis and its applications branch off with the use of analytical methods, some of them close to works on register allocation or sorting networks. Variance analysis is related to the domain of the so-called "q-analogs" an active domain of combinatorial theory.
In this brief survey, we hope to convince the reader that the analysis of even the simplest algorithms (sorted and unsorted lists) in a dynamic context is a very rich field with a host of interesting combinatorial as well as analytic implications.

PRELIMINARY DEFINITIONS

We consider in the sequel operations of the following three types : Insertions (I) ; Deletions (D) ; Queries (Q).

Let $\mathcal{U}$ be the set of all possible keys -the "universe" of keys-. A sequence of operations of length n relative to $\mathcal{U}$ is a sequence of the type $\omega_1(k_1)\omega_2(k_2) \ldots \omega_n(k_n)$ where $\omega_i \in \{I,D,Q\}$ and $k_i \in \mathcal{U}$. Various limitations can be imposed on the admissible sequences of operations, giving rise to different data types ; the types we consider here are :
a) dictionaries (DICT) where operations and keys at any stage are unrestricted ;
b) dictionaries without queries whose sequences of operations are also isomorphic to linear lists (LL) : here the ω's can only be I or D ;
c) priority queues (PQ) where operations can be either I or D but a deletion only operates at each stage on the smallest key present in the structure.

To be meaningful a sequence of operations together with all its prefixes must contain more I's than D's (no deletion on an empty file can occur and we shall assume this natural condition to be satisfied). The foregoing analyses will be performed with respect to various sequences of operations each comprising as many I's as D's (thus preventing the analysis to be biased towards insertions). In each case we let Ω_n denote the set of all such sequences of length n, and let H_n denote card Ω_n.
We are interested in the behaviour of data structures w.r.t. to either time or storage, under sequences of operations in Ω_n. To each particular structure is associated the set of its unitary costs CI_k, CD_k, CQ_k where $C\omega_k$ is the cost of operation ω performed on a file of size k[†].

[†] In applications, the cost of an operation is not only a function of the size of the file but also of the particular organization of keys resulting from past operations. It can however be proved that such coefficients $C\omega_k$ exist for structures that stay "random" under deletions [4]. The $C\omega_k$ are then themselves computable as certain averages over configurations of the structure.

I - FINITE FILES

When the universe of keys is finite, say with cardinality N, the set of allowable input sequences of length n is also finite and we have three different types of statistics PQ/N/, LL/N/, DICT/N/. To estimate integrated costs we first need to count the number of possible request sequences, card $\Omega_n = H_n$. The reader may try to convince himself that

$$H_{2n}^{PQ/2/} = 3^n - 3^{n-1} \quad \text{and} \quad H_n^{DICT/3/} = \frac{1}{8}\,6^n + \frac{3}{8}\,4^n + \frac{3}{8}\,2^n .$$

Theorem 1^A : [*Generating functions for sequences of operations*]

a) *For* DICT/N/, *i-e dictionaries over a universe of* N *keys*

$$\sum H_n \frac{z^n}{n!} = \left(\frac{e^{2z}+1}{2}\right)^N \quad \text{and} \quad \sum H_n\, z^n = \sum_j \frac{\binom{N}{j}}{1-2jz} \; ;$$

b) *For* LL/N/, *i-e dictionaries without queries over* N *keys*

$$\sum H_n \frac{z^n}{n!} = \left(\frac{e^{z}+e^{-z}}{2}\right)^N \quad \text{and} \quad \sum H_n\, z^n = \sum_j \frac{\binom{N}{j}}{1-(N-2j)z} \; ;$$

c) *For* PQ/N/, *i-e priority queues*

$$\sum H_n\, z^n = \frac{He_N(z)}{He_{N+1}(z)}$$ *where* $He_m(z)$ *is the* m-th *Hermite polynomial* : $He_m(z) = \sum_r \frac{m!}{r!(m)2r)!} \left(-\frac{z^2}{2}\right)^r .$ □

Let $H_{o,k,n}$ denote the number of extended sequences of operations resulting in a structure of size k i-e comprising k more I's than D's (thus $H_n = H_{o,o,n}$). Then

Theorem 1^B : [*Generating functions for extended sequences of operations*]

a) *For* DICT/N/ : $\sum H_{o,k,n} \frac{z^n}{n!} = \binom{N}{k} e^{Nz} \cosh^{N-k} z \sinh^k z$;

b) *For* LL/N/ : $\sum H_{o,k,n} \frac{z^n}{n!} = \binom{N}{k} \cosh^{N-k} z \sinh^k z$;

c) *For* PQ/N/ : $\sum H_{o,k,n}\, z^n = \frac{N!}{(N-k)!} \frac{He_{N-k}(z)}{He_{N+1}(z)}$. □

These results rely on the use of the continued fraction theorem of Flajolet [2] ; the three finite fractions relative to the generating functions $\sum H_n z^n$ of theorem 1^A are

DICT/N/ :

$$\cfrac{1}{1-Nz-\cfrac{N.1z^2}{1-Nz-\cfrac{(N-1)2z^2}{\cdots}}} \; ;$$

LL/N/ :

$$\cfrac{1}{1-\cfrac{N.1z^2}{1-\cfrac{(N-1)2z^2}{\cdots}}} \; ;$$

PQ/N/ :

$$\cfrac{1}{1-\cfrac{Nz}{1-\cfrac{(N-1)z}{\cdots}}}$$

The expressions given above yield expressions for the $H_{o,k,n}$, H_n... by combinatorial sums. For example :

$$H_n^{DICT/N/} = 2^{n-N}\left[N^n + \binom{N}{1}(N-1)^n + \binom{N}{2}(N-2)^n + \ldots\right]$$

Given a data structure with unitary costs CI_k, CD_k, CQ_k, we wish to compute the corresponding integrated costs $\overline{K}_n = K_n / H_n$ where $K_n = \sum_{u \in \Omega_n} \mathrm{cost}(u)$. We assume for ease of exposition that $CI_k = CD_k = CQ_k = C_k$.

Theorem 2^A : [*Integrated costs for priority queues over* N *elements*]
For priority queues, the generating function of total costs $K(z) = \sum K_n z^n$ *is a rational fraction given by*

$$K(z) = \frac{N!}{\left(He_{N+1}(z)\right)^2} \sum_k C_{N-k} \frac{z^{2N-2k}}{(N-k)!} \left(He_k(z)\right)^2 . \qquad ☐$$

Similar rational expressions can also be given for linear lists and dictionaries. In these last two cases, it may also prove convenient to operate with exponential generating functions taking

$$\hat{K}(z)' = \sum K_n \frac{z^{n+1}}{(n+1)!} \quad \text{and} \quad C(u) = \sum_k C_k \binom{N}{k} u^k$$

as generating functions of integrated costs and unitary costs.

Theorem 2^B : [*Integral transforms for integrated costs*] : *The generating functions of unitary and integrated costs are related by* :

$$\mathrm{DICT}/N/ : \hat{K}(z) = \frac{\left(e^{2z}+1\right)^N}{2} L^{/N/}[C(u)] \quad ; \quad \mathrm{LL}/N/ : \hat{K}(z) = \cosh^N z \, L^{/N/}[C(u)]$$

where the linear transform $L^{/N/}$ *is* :

$$L^{/N/}[C(u)] = 2 \int_0^{\tanh^2 z/2} \frac{C(u)}{(1+u)^N (1-u)} \frac{du}{\sqrt{(1+u)^2 \tanh^2 z - 4u}} . \qquad ☐$$

As in the other cases discussed in [3], this last theorem reduces the determination of integrated costs to the evaluation of integrals of elementary functions relayed by the computation of Taylor expansions. Both theorem 2^A and theorem 2^B permit easy numerical tabulations of the integrated costs of an arbitrary structure.

As an application these results enable us to compare two priority queue organizations A and B with corresponding integrated costs $\overline{K}_n^A$ and $\overline{K}_n^B$ under the PQ/N/ statistics. It can be shown that the quantities $\overline{K}_n^A/n$ which represent the cost of a random operation in a random operation sequence of length n tend, as n gets large, to a constant $\overline{K}_\infty^A$ which, in a way, represents the "steady-state" cost of an operation in A under the statistics PQ/N/.

Theorem 2^C : [*Steady state cost under* PQ/N/]
The steady state cost of an operation of algorithm A *under the statistics* PQ/N/ *is* $\overline{K}_\infty^A = \sum_{0 \le k \le N} C_k^A \pi_k$ *where* :

$$\pi_k = \frac{N!}{\alpha \hat{He}_N(\alpha) \hat{He}'_{N+1}(\alpha)} \frac{\alpha^k}{k!} \hat{He}^2_{N-k}(\alpha) ;$$

here $\hat{He}(z) = He(\sqrt{z})$ *and* α *the (positive real) root of* $\hat{He}_{N+1}(z)$ *with smallest modulus*. ☐

Notice that the coefficient π_k is nothing but the limiting probability that an operation takes place at level k in the course of a sequence of length 2n (see in section 2 the related notion of profile).
Theorem 2^C gives a numerical algorithm that suffices in all cases to compare two different data structures.

A treatment of linear lists and dictionaries could be given along very similar lines.

II - PROFILES

We now consider the case where the universe of keys U is infinite. The sampling technique introduced by Françon [4] amounts to selecting one representant in each order-isomorphism class of input sequences. Canonical representants called histories are defined by retaining at each stage only the order of the key operated upon w.r.t. keys already in the structure. For example , for priority queue data type, a history is $I_0I_0I_2D_0D_0D_0$ which represents any sequence of operations obeying the pattern : insert a key ; insert another key smaller than the first one (it then has rank 0 w.r.t. the first key), insert another key larger than the two previous ones (it then has rank 2 w.r.t. to these two keys), delete minimum... ; such a history is a canonical represent of say I(2.7) I(0.5) I(3.1) D_{min} D_{min} D_{min}.

We use the same notations as before : H_n represents the number of histories (from 0 to 0) of length n, K_n represents the total cost of all histories of length n and $\overline{K}_n=K_n/H_n$ is the integrated cost of sequences of length n.

It has been shown that priority queue histories lead to the continued fraction

$$\sum_{n\geq o} H_n z^n = \cfrac{1}{1-\cfrac{1z^2}{1-\cfrac{2z^2}{1-\cfrac{3z^2}{\cdots}}}}$$

and enumerations again involve the Hermite polynomials. From it follows the simple enumeration $H_{2n} = 1.3.5...(2n-1) = n?$.

We recall the following result which has been used to analyze a variety of simple structures :

Theorem 3^A : [*Integrated cost of priority queues*] *The generating function* $\tilde{K}(z) = \sum_{n\geq o} K_{2n} \frac{z^n}{n!}$ *of the integrated cost of priority queues is related to the generating function of individual costs* $C(u) = \sum_{k\geq o} (CI_k+CD_{k+1})u^{k+1}$ *by*

$$\tilde{K}(z) = \frac{1}{\sqrt{1-2z}}\, C\left(\frac{z}{1-z}\right) . \qquad \square$$

Setting $C_k = CI_k + CD_{k+1}$, the above expression is equivalent to combinatorial expressions of integrated costs :

$$K_{2n} = \sum_{m\leq k\leq n} C_k\, 2^{m-n} \binom{n-m-1}{k-1} \frac{\binom{2m}{m}}{\binom{2n}{n}} \quad \text{whence the } \overline{K}_{2n} = K_{2n}/H_{2n} .$$

These expressions are already difficult to handle when $C_k = 1+\frac{1}{2}+\frac{1}{3}\cdots+\frac{1}{k}$. The following result gives a very simple way of computing the first terms in the asymptotic expansion of $\overline{K}_{2n}$:

Theorem 3^B : *For smoothly growing individual costs* C_k, *the integrated costs satisfy*

$$\overline{K}_{2n} = n \int_0^{1/2} C_{\lfloor n\phi\rfloor} \frac{d\phi}{\sqrt{1-2\phi}} \left(1+O\left(\frac{1}{n}\right)\right) . \qquad \square$$

Theorem 3 tells us that weighting corresponding to the passage of individual costs to integrated costs can be described asymptotically by means of an integral form (Previous ways of computing integrated costs required a recourse to generating functions). The proof is achieved by considering the quantities $\pi_{k,n}$ which are the multipliers of C_k in the expression of $\overline{K}_{2n}$; $\pi_{k,n}$ represents the probability of an operation being performed on a file of size k in the course of a random history of length 2n. It is shown that the $\pi_{k,n}$ have a limit distribution with density $\frac{1}{\sqrt{1-2\phi}}$.

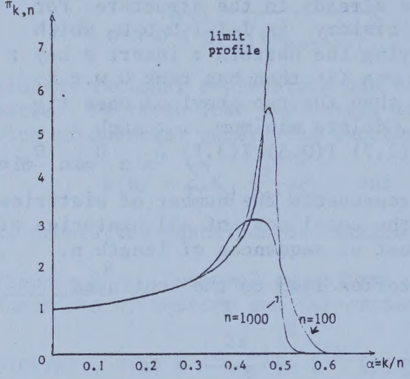


Figure 1 : The limit distribution of $\pi_{k,n}$ and the two cases n=100, n=1000.

The emergence of the limit distribution of the $\pi_{k,n}$ is demonstrated on figure 1 in the cases when n=100 and n=1000. With theorem 3 we readily rederive the values of $\overline{K}_{2n}$ corresponding to individual costs of a simple form :

$$C_k = 1 \Rightarrow \overline{K}_{2n} \sim n \ ;$$

$$C_k = k \Rightarrow \overline{K}_{2n} \sim \frac{n^2}{3} \ ;$$

$$C_k = 1 + \frac{1}{2} + \ldots + \frac{1}{k} \Rightarrow \overline{K}_{2n} \sim n \log n$$

The main interest of theorem 3, besides its simplicity, is to make possible analysis of rather complex structures. We study the case of binomial queues, one of the most efficient administration policies for priority queues, which was discovered by Vuillemin [6].
The structure is based upon a decomposition into blocks of size 1,2,4,8,... . The analysis of individual costs in a good implementation of binomial queues has been performed by Brown [1] who showed :

Theorem : [*Individual costs for binomial queues*]
Let $k = \sum b_i 2^i$ *be the binary decomposition of* n (b_i = 0 or 1). *Then the undividual costs of binomial queues are* $CI_k = v_2(k)$ *and* $CD_{k+1} = \sigma(k) - v_2(k)$ *where* $v_2(k) = \min\{i \mid b_i \neq 0\}$ *is the number of trailing zeros in the binary representation of* k *and* $\sigma(k)$ *is an arithmetical function given by*

$$\sigma(k) = \frac{1}{k} \sum_i i\, b_i 2^i . \qquad \square$$

We thus have in this case $C_K = CI_k + CD_{k+1} = \sigma(k)$. Although an expression for $C(u) = \sum c_k u^k$ is available as a Lambert series, it proves rather difficult to work with. Accordingly, $\sigma(k)$ has a rather erratic behaviour reflecting the steep threshold effects for values of k that are powers of 2. The approach through theorem 3 proves extremely useful in this context. We first have :

Theorem 4^A : [*Analytic expression of* $\sigma(k)$] *The function* $\sigma(2p)$ *is expressible as*

$$\sigma(2p) = \log_2 p + G(\log_2 p)$$

where G *is a bounded periodic function with period* 1, *with Fourier coefficients :*

$$g_0 = \frac{1}{2} + \frac{\gamma-1}{\log 2} \quad ; \quad g_k = \frac{1}{\log 2} \; \frac{\zeta(1+\chi_k)}{1+\chi_k} \quad \left(\chi_k = \frac{2ik\pi}{\log 2}\right) .^{\dagger} \qquad \square$$

Function G is highly discontinuous. The above expression can however be combined with theorem 3 to yield.

Theorem 4^B : [*Integrated cost of binomial queues*] *The integrated cost of* 2n *operations on a binomial queue is* $\overline{K}_{2n} = n \log_2 n + n\, P(\log_2 n) + o(n)$ *where* P *is a continuous function with period* 1, *whose Fourier coefficients* p_k *are*

$$p_0 = \frac{1}{2} + \frac{\gamma-3}{\log 2} \quad ; \quad p_k = \frac{1}{\log 2} \; \frac{\Gamma(1+\chi_k)^2}{\Gamma(2(1+\chi_k))} \; \frac{\zeta(1+\chi_k)}{1+\chi_k} . \qquad \square$$

These estimates are in good agreement with the actual integrated costs. On figure 1 below we have plotted the values of

$Q(n) = \frac{\overline{K}_{2n}}{n} - \log_2 n$, and this last term rapidly conforms to $P(\log_2 n)$.

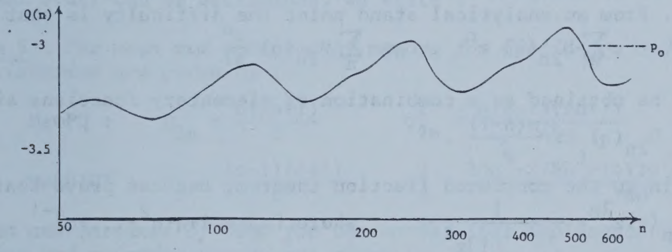


Figure 2 : The quasi-periodic term in the integrated cost of binomial queues.

Similar treatments could be given for dictionaries and linear lists.

III - VARIANCES

We have shown in the previous two sections ways of estimating costs either exactly or asymptotically in various contexts of requests. We here show that the continued fraction approach can be used to also attain results about typicality of the average case. More precisely, we consider five problems hereafter called MemPQ, MemDICT, TimePQ-SL, TimePQ-UL, TimeDICT-SL. Here Mem stands for memory, PQ and DICT retain their meaning as priority queue and dictionary types, SL is the sorted list and UL the unsorted list structure. So, for instance TimePQ-SL refers to the time cost of the sorted list implementation of priority queues, MemPQ refers to the total storage occupation measured in the number of keys stored under the statistics of priority queues histories... . These various problems are studied in the context of an infinite set of keys as in section 2. We shall present the line of attack on TimePQ-SL, i-e number of comparison (*comp*) in the sorted list implementation of priority queues under sequences of operations. Given a priority queue history, the number of resulting comparisons in the sorted list implementation is readily determined : when an insertion operates with rank r, the number of comparisons performed to place the element in the structure is plainly r+1 ; a deletion requires no comparison

---

† Γ and ζ refer resp. to the Euler gamma function and the Riemann zeta function ; γ is the Euler constant.

at all since the minimum element always occurs first in the structure. The number of comparisons associated with $h=I_0 I_0 I_2 I_1 I_4 D_{min} D_{min} I_2 D_{min} D_{min} D_{min} D_{min}$ is $comp(h) = (0+1) + (0+1) + (2+1) + (1+1) + (4+1) + (2+1) = 13$.

Let $H_{n,c}$ be the number of priority queue histories of length n leading to c comparisons in SL. We are interested in the distribution of *comp*, and more specifically to its mean and variance.

From results recalled in section 2, we know that $\sum_c H_{2n,c} = 1.3.5...(2n-1) = n?$, representing the total number of priority queue histories.

The $H_{2n,c}$ have a number of interesting combinatorial properties :

$$H_{2n,c} \neq 0 \text{ iff } n \leq c \leq \frac{n(n+1)}{2} \quad ; \quad H_{2n,n} = \frac{1}{n+1}\binom{2n}{n} \quad ; \quad H_{2n,\frac{n(n+1)}{2}} = 1 .$$

In the sequel we shall work with the polynomials $H_{2n}(q) = q^{-n} \sum_c H_{2n,c} q^c$.

The parameter *comp* belongs to the category of cumulative parameters in the analysis of algorithms. Such parameters have quadratic worst case and occur for instance in the analysis of insertion sort, bubble sort, tree sort, quisksort... . From an analytical stand point the difficulty is that a series like

$$\sum_n H_{2n}(q)\, z^n \quad \text{of} \quad \sum_n H_{2n}(q) \frac{z^n}{n!} \ldots ,$$

can in no way be obtained as a combination of elementary functions since the degree of $H_{2n}(q)$ is $\frac{n(n-1)}{2}$.

Appealing again to the continued fraction theorem, one can prove that :

$$H(z;q) = \sum H_{2n}(q) z^{2n} = \cfrac{1}{1 - \cfrac{[1]z^2}{1 - \cfrac{[2]z^2}{1 - \cfrac{[3]z^2}{\cdots}}}} \qquad \text{where } [s] = 1+q+q^2+\ldots+q^{s-1} .$$

This continued fraction reduces to the continued fraction of section 2 when q=1 of which it is said to be a q-analog.

The mean and variance of *comp* over histories of length 2n are given by :

$$\mu_{2n} = \frac{1}{n?}\left[\frac{dH_{2n}}{dq}(q)\right]_{q=1} + n \quad ; \quad \sigma^2_{2n} = \frac{1}{n?}\left[\frac{d^2}{dq^2} H_{2n}(q)\right]_{q=1} - \mu^2_{2n} + \mu_{2n},$$

but it is to be noted that H(z;q) together with its derivates at q=1 strongly diverge.

The convergents of H(z;q) are finite fractions which can be expressed as $\frac{P_r(z;q)}{Q_r(z;q)}$ with P_r and Q_r polynomials in z and q. The denominator polynomials Q_r satisfy a classical recurrence relation ; they are related to q-analogs of the Hermite polynomials.

A Eulerian generating function of the reciprocal polynomials of Q_r defined by : $K(t,z;q) = \sum_{r \geq o} Q_{r-1}(\frac{1}{z}|q) \frac{z^r t^r}{[1][2]\ldots[r]}$, (this reduces to $e^{-(t^2/2)+tz}$ when q=1) satisfies the difference equation $\frac{K(qt,z;q)-K(t,z;q)}{(q-1)t} = (z-t)K(t,z;q)$.

This equation reduces to a classical differential equation when $q=1$; it can be "solved" expressing K either as an infinite product, or in terms of the q-exponential. It suffices here to notice that this functional equation makes it possible to determine the values of

$$\left[\frac{d}{dq}\ Q_r(z;q)\right]_{q=1} , \left[\frac{d^2}{dq^2}\ Q_r(z;q)\right]_{q=1} \cdots$$

after a rather difficult computation. Using orthogonality relations that hold between the Q's and the H's further allows determination of

$$\left[\frac{d}{dq}\ H_{2n}(q)\right]_{q=1} , \left[\frac{d^2}{dq^2}\ H_{2n}(q)\right]_{q=1} \cdots$$

and we have the very simple result :

$$a)\ \mu_{2n} = \frac{n(n+5)}{6} \quad ; \quad b)\ \sigma^2_{2n} = \frac{n(n-1)(n+3)}{45} .$$

These results are related to combinatorial results by Touchard and Riordan relative to chord intersection problems. We have here a unified approach which leads to a solution of all the problems mentioned above. Also moments of higher order can be determined. We state :

Theorem 5 : *The mean and variance of memory for dictionary and priority queue histories are given by :*

$$\text{MemPQ :} \quad \mu_{2n} = \frac{n(2n+1)}{3} \qquad \sigma^2_{2n} = \frac{n(n-1)(2n+1)}{45}$$

$$\text{MemDICT :} \quad \mu_n = \frac{(n-1)(n+1)}{6} \qquad \sigma^2_n = \frac{20n^4-278n^3+1472n^2-3073n+2047}{180} .$$

The mean and variance of time for the sorted list implementation of dictionaries and priority queues are given by

$$\text{TimePQ-SL :} \quad \mu_{2n} = \frac{n(n+5)}{6} \qquad \sigma^2_{2n} = \frac{n(n-1)(n+3)}{45}$$

$$\text{TimeDICT-SL :} \quad \mu_n = \frac{(n-1)(n-2)}{12} \qquad \sigma^2_n = \frac{(n-2)(2n^2+11n-1)}{360} . \qquad \square$$

For instance the proof of MemPQ uses another q-analog of Hermite polynomials whose associated generating function satisfies the difference-differential equation

$$\frac{\partial K(t,z;q)}{\partial t} = zK(t,z;q) - qK(qt,z;q).$$

The problem TimePQ.UL of the time complexity for unsorted lists actually reduces to MemPQ. Figure 3 gives the comparative distributions in number of comparisons of sorted versus unsorted lists under the PQ statistics when n=25.

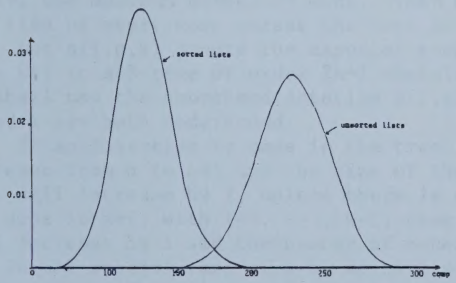


Figure 3 : Distribution of costs for sorted vs. unsorted implementations of priority queues.

REFERENCES

[1] M.R. Brown,
Implementation and analysis of binomial queues algorithms,
SIAM J. Comput., vol. 7, n° 3 (1978), pp. 298-319.

[2] P. Flajolet,
Combinatorial aspects of continued fractions,
Discrete Maths, (1980), to appear.

[3] P. Flajolet, J. Françon, J. Vuillemin,
Sequence of Operations Analysis of Dynamic Data Structures,
Journal of Algorithms, 1, (1980), pp. 111-141.

[4] J. Françon,
Combinatoire des Structures de Données
Thèse, Université Louis Pasteur, Strasbourg (1979).

[5] D. Knuth,
The Art of Computer Programming,
Vol. 3, (1973).

[6] J. Vuillemin,
A Data Structure for Manipulating Priority Queues,
Comm. ACM 21, n° 4, pp. 309-315.

Mailing address : Ph. FLAJOLET, INRIA, 78150 ROCQUENCOURT (FRANCE).

AVERAGE PERFORMANCE OF THE B-TREE

WILLIAM E. WRIGHT
Department of Computer Science
Southern Illinois University at Carbondale
Carbondale, Illinois

ABSTRACT

Average values for the following B-tree performance measures are derived: storage utilization, tree height, number of splits per insertion, total number of nodes, number of nodes of each size (where the size of a node is the number of keys currently contained in the node), and number of accesses required for a retrieval or an insertion.

INTRODUCTION

B-trees or variations of B-trees are popular structures for the organization of indexed files in secondary memory. B-trees were first introduced by Bayer and McCreight in [1], and have subsequently been discussed in [2-7]. The following bounds on B-tree performance have been derived in the past:

a. The minimum storage utilization is 50%, excluding the root.

b. The maximum tree height is $1 + \log_{k+1}\frac{n+1}{2}$, where 2k is the maximum number of keys per node and n is the number of keys in the index.

c. The minimum height of the tree is $\log_{2k+1}(m+1)$.

d. The expected number of splits per insertion is less than 1/k.

The purpose of this paper is to present derivations of *average* values for these and other performance measures. As expected, the averages are significantly better than the worst-case bounds. Averages are derived for the storage utilization, tree height, number of splits per insertion, total number of nodes, number of nodes of each size (where the size of a node is the number of keys currently contained in the node), and number of accesses required for a retrieval or an insertion.

RELATIVE NUMBER OF NODES OF EACH SIZE

The analysis of the average performance of B-trees is based largely on a determination of the average number of nodes of each size. Let $k \geq 1$ be the minimum number of keys in a node (except the root), 2k the maximum, and m=2k+1 the maximum number of sons. Then m is the order of the tree, and the size of every node except the root is between k and 2k, inclusive.

Let A(i,n,k) denote the expected number of nodes (except the root) of size k+i in a B-tree of order 2k+1 containing n keys, where i=0, ···, k. We shall use the shorthand notation A(i,n) if k is understood, or A(i) if n and k are both understood.

If an insertion is made in the tree, then the number of keys will increase from n to n+1 and the size of the node in which the insertion is made will increase by 1, unless there is a split. If the original size of the node is k+i, with i=0, ···, k-1, then the number of nodes of size k+i will decrease by 1 and the number of nodes of size k+i+1 will increase by 1. In the special case of i=k, we have a split, so that the number of nodes of size 2k will decrease by 1 and the number of nodes of size k will increase by *2*. In addition, a split will cause a subsequent insertion into the father of the splitting node.

The number of gaps into which an insertion can be made is n+1, if the number of keys in the tree is n. If the sequence of insertions is random, then each gap is equally likely. (Insertions and deletions are quite analogous in this regard, except that with deletions the search key must be in the tree, and the number of equally likely possibilities is n instead of n+1.)

It is important to note that while an insertion into a B-tree must be made at the bottom level of the tree, higher levels of the tree may receive insertions due to node splitting. Moreover, the average relative number of nodes of each size is the same for every level (except the root). Hence the derivation at this point can ignore level number without loss of generality.

We now assume that the tree has reached an equilibrium state, in the sense that the relative number of nodes of each size is approximately constant under the influence of additional insertions. Therefore we must have

$$\frac{A(i,n)}{A(j,n)} = \frac{A(i,n+1)}{A(j,n+1)} \quad , \ i,j=0, \cdots, k, \text{ or}$$

$$\frac{A(i,n)}{A(j,n)} = \frac{A(i,n+1) - A(i,n)}{A(j,n+1) - A(j,n)} \quad , \ i,j=0, \cdots, k \tag{1}$$

Since the number of gaps in a node of size k+i is k+i+1, the probability of making an insertion into a node of size k+i is

$$\frac{k+i+1}{n+1} A(i,n)$$

Hence we have

$$A(i,n+1) = A(i,n) + \frac{1}{n+1}((k+i)A(i-1,n) - (k+i+1)A(i,n)), i=1,\cdots,k \tag{2}$$

$$A(0,n+1) = A(0,n) + \frac{1}{n+1}((2k+1)A(k,n)\cdot 2 - (k+1)A(0,n)) \tag{3}$$

Combining (1) - (3) we get

$$\frac{A(i,n)}{A(j,n)} = \frac{(k+i)A(i-1,n)-(k+i+1)A(i,n)}{(k+j)A(j-1,n)-(k+j+1)A(j,n)} \quad , \ i,j=1, \cdots, k \tag{4}$$

$$\frac{A(0,n)}{A(j,n)} = \frac{(2k+1)A(k,n)\cdot 2-(k+1)A(0,n)}{(k+j)A(j-1,n)-(k+j+1)A(j,n)} \quad , \ j=1, \cdots, k \tag{5}$$

Setting j to i+1 in (4), we get

$$\frac{A(i,n)}{A(i+1,n)} = \frac{(k+i)A(i-1,n)-(k+i+1)A(i,n)}{(k+i+1)A(i,n)-(k+i+2)A(i+1,n)} \quad , \ i=1, \cdots, k-1, \text{ or}$$

(dropping n)

$$(k+i)A(i-1)A(i+1) = (k+i+1)A^2(i)-A(i)A(i+1) \quad , \ i=1, \cdots, k-1, \text{ or}$$

$$A(i-1)A(i+1)-A^2(i) = \frac{1}{k+i}(A^2(i)-A(i)A(i+1)) \quad , \ i=1, \cdots, k-1 \tag{6}$$

Setting j to 1 in (5), we get

$$\frac{A(0,n)}{A(1,n)} = \frac{(2k+1)A(k,n)\cdot 2-(k+1)A(0,n)}{(k+1)A(0,n)-(k+2)A(1,n)} \quad , \text{ or (dropping n)}$$

$$(4k+2)A(1)A(k) = (k+1)A^2(0)-A(0)A(1) \tag{7}$$

Using (6) and (7), it can be shown by contradiction that

$$A(i) > A(i+1), \ i=0, \cdots, k-1 \tag{8}$$

Using this result, it follows from (6) that

$$\lim_{k\to\infty} \frac{A(i-1)\,A(i+1)}{A^2(i)} = 1 \qquad , i=1, \cdots, k-1$$

Therefore for large k it is approximately true that

$$A(i-1)\,A(i+1) = A^2(i),\ i=1, \cdots, k-1, \text{ or}$$

$$A(i) = \frac{A^2(i+1)}{A(i+2)},\ i=0, \cdots, k-2$$

Applying this equation repeatedly as i goes from k-2 to 0, we get

$$A(i) = \frac{A^{k-i}(k-1)}{A^{k-i-1}(k)},\ i=0, \cdots, k-1 \tag{9}$$

$$A(i) = A(i+1)\,\frac{A(k-1)}{A(k)},\ i=0, \cdots, k-1 \tag{10}$$

Returning to (7) and rearranging, we have

$$4k\,A(1)\,A(k) - k\,A^2(0) = A^2(0) - 2\,A(1)\,A(k) - A(0)\,A(1) \qquad , \text{ or}$$

$$\frac{4\,A(1)\,A(k)}{A^2(0)} - 1 = \frac{1}{k}\left(\frac{A^2(0)}{A^2(0)} - \frac{2A(1)\,A(k)}{A^2(0)} - \frac{A(0)\,A(1)}{A^2(0)}\right) \tag{11}$$

It follows from (8) that the right side of Equation (11) approaches 0 as k approaches infinity, therefore for large k it is approximately true that

$$4\,A(1)\,A(k) = A^2(0) \tag{12}$$

Using (9) with i=0 and i=1 gives

$$A(0) = A^k(k-1)/A^{k-1}(k) \text{ and}$$

$$A(1) = A^{k-1}(k-1)/A^{k-2}(k)$$

Substituting this in (12) gives

$$4\,A^{k-1}(k-1)/A^{k-2}(k) \cdot A(k) = A^{2k}(k-1)/A^{2k-2}(k) \text{ or}$$

$$4\,A^{k+1}(k) = A^{k+1}(k-1) \text{ or}$$

$$A(k-1) = 4^{\frac{1}{k+1}}\,A(k)$$

Substituting this in (9) yields

$$A(i) = \frac{4^{\frac{k-i}{k+1}}\,A^{k-i}(k)}{A^{k-i-1}(k)} \qquad , i=0, \cdots, k-2, \text{ or}$$

$$A(i) = 4^{\frac{k-i}{k+1}}\,A(k),\ i=0, \cdots, k \tag{13}$$

Table 1 gives relative values of A(i) for k=4, assuming A(k)=1.

INSERTION PROBABILITIES AND SPLITTING

From (13) it is possible to derive formulas for several average performance measures for B-trees. We shall use the symbol α to denote the expression $\sqrt[k+1]{.25}$, which is needed many times.

Let S(k) denote the number of gaps in the tree (excluding the root) where an insertion can be made. Then

$$S(k) = \sum_{i=0}^{k} (k+i+1)\ A(i)$$

$$= \sum_{i=0}^{k} (k+i+1)\ 4^{\frac{k-i}{k+1}}\ A(k)$$

$$= \frac{\alpha}{(1-\alpha)^2}\ (3 + (2k-1)\ (1-\alpha))\ A(k) \tag{14}$$

(Details of the derivation are omitted.)

Let p(i,k) denote the probability that an insertion into the tree is made into a node of size k+i, where i=0, ···, k. Then

$$p(i,k) = (k+i+1)\ A(i)/S(k),\ i=0,\ \cdots,\ k$$

Substituting from (13) and (14), this gives

$$p(i,k) = \frac{4(k+i+1)\ \alpha^i\ (1-\alpha)^2}{3 + (2k-1)\ (1-\alpha)},\ i=0,\ \cdots,\ k \tag{15}$$

Table 2 contains values of p(i,k) for k=4.

The probability t(k) that an insertion into the tree will result in a split is equal to the probability that the insertion is made into a node which is full, i.e., p(k,k). Thus

$$t(k) = p(k,k) \text{ or}$$

$$t(k) = \frac{4(2k+1)\ \alpha^k\ (1-\alpha)^2}{3 + (2k-1)\ (1-\alpha)} \tag{16}$$

Table 3 contains values of t(k) for several values of k. As suggested by the table, the limit of t(k) as k approaches infinity is 0.

Equation (16) can be used to determine the average number of splits (u(k), say) during an insertion into a B-tree. Recall that the insertion of a single key into the tree may result in more than one split since each split requires a subsequent insertion into the father of the splitting node. If the number of levels in the tree (excluding the root) is b, then the probability of an insertion causing exactly i splits, i=0, 1, ···, b, is $t^i(k)\ (1-t(k))$. Therefore,

$$u(k) = \sum_{i=0}^{b} i\ t^i(k)\ (1-t(k))$$

$$\simeq (1-t(k)) \sum_{i=0}^{\infty} i\ t^i(k) \text{ or}$$

$$u(k) \simeq \frac{t(k)}{1-t(k)} \tag{17}$$

We can easily substitute into (17) from (16), but we shall instead be content to look at the limit of u(k) as k approaches infinity.

Specifically, we wish to compare (in the limit) u(k) with its upper bound 1/k. Therefore, let

$$g(k) = u(k)/(1/k) = k\ u(k)$$

Then

$$\lim_{k\to\infty} g(k) = \lim \frac{k\ t(k)}{1-t(k)}$$

$$= \lim k\ t(k)$$

$$= \frac{2\ (\ln 4)^2}{3 + 2\ \ln 4}$$

$$\simeq .66584$$

Thus for large k the expected number of splits per insertion in the tree is approximately $.66584/k \simeq \dfrac{1}{1.502\cdot k}$, or two-thirds as large as the upper bound.

STORAGE UTILIZATION

The average storage utilization in a B-tree can be derived from (13) in a straight-forward fashion. A node of size k+i is storing k+i keys, where i=0, $\cdots$, k. Each node has a capacity of 2k keys. Therefore the average storage utilization v(k) is

$$v(k) = \frac{\sum\limits_{i=0}^{k} (k+i)\ A(i)}{2k \sum\limits_{i=0}^{k} A(i)}$$

$$= \frac{k \sum\limits_{i=0}^{k} 4^{\frac{k-i}{k+1}} + \sum\limits_{i=0}^{k} i\ 4^{\frac{k-i}{k+1}}}{2k \sum\limits_{i=0}^{k} 4^{\frac{k-i}{k+1}}}$$

$$= \frac{\dfrac{3k\alpha}{1-\alpha} + \dfrac{4^{\frac{k}{k+1}}}{(1-\alpha)^2}\ (\alpha-.25-.25k(1-\alpha))}{\dfrac{6k\alpha}{1-\alpha}} \quad \text{or}$$

$$v(k) = \frac{1}{3} + \frac{2}{3}\ \frac{\alpha-.25}{k(1-\alpha)} \tag{18}$$

Table 4 contains values of v(k) for several values of k.

Using (18) we can determine that

$$\lim_{k\to\infty} v(k) = \frac{1}{3} + \frac{2}{3}\ \frac{1-.25}{\ln 4}$$

$$= \frac{1}{3} + \frac{1}{2\ \ln 4}$$

$$\simeq .69401 \tag{19}$$

From Table 4 it appears that v(k) is an increasing function of k, for $k \geq 1$. This is in fact true, although the proof is quite lengthy and so is not given here. Therefore $\frac{1}{3} + \frac{1}{2\ \ln 4} \simeq .69401$ is an upper bound for v(k), and

$v(1) = \frac{2}{3} \simeq .66666$ is a lower bound. Therefore $v(k)$ lies in the small range from .66666 to .69401. Therefore the average storage utilization in a B-tree is slightly less than 70%.

TREE HEIGHT AND ABSOLUTE NUMBER OF NODES

Equations (18) and (19) can be used in conjunction with parameter n to determine the average height of a B-tree. Each node (except the root) contains an average of 2k v(k) keys and 2k v(k)+1 sons. Assuming the root is half full, it contains k keys and k+1 sons. Therefore the average height h(k,n) of a tree containing n keys is given approximately by

$$h(k,n) = 1 + \log_{2k\ v(k)+1} \frac{n}{k+1}$$

$$\simeq 1 + \log_{1.38\ k+1} \frac{n}{k+1} \qquad (20)$$

Table 5 contains several values of h(k,n) for k=50. Equation (20) can be simplified slightly by approximating the root to have the same size as the other nodes. Then we get

$$h(k,n) = \log_{2k\ v(k)+1} n \text{ or}$$

$$h(k,n) \simeq \log_{1.38k+1} n \qquad (21)$$

It is important to note that (20) and (21) also represent, in addition to the average tree height, the average number of accesses required to search a B-tree.

Parameter n can also be used with Equation (13) to determine absolute values for A(0), ···, A(k). Again assuming k keys in the root, we have

$$n-k = \sum_{i=0}^{k} (k+i)\ A(i)$$

$$= \left(\frac{3k\alpha}{1-\alpha} + \frac{4^{\frac{k}{k+1}}}{(1-\alpha)^2} \quad (\alpha-.25-.25k(1-\alpha))\right)\ A(k)$$

Therefore we have

$$A(k) = \frac{(1-\alpha)^2\ \ (n-k)}{\alpha(3+(2k-4)(1-\alpha))} \qquad (22)$$

Substituting (22) into (13) we get

$$A(i,n,k) = \frac{4^{\frac{k-i}{k+1}}\ (1-\alpha)^2\ \ (n-k)}{\alpha(3+(2k-4)\ (1-\alpha))}\ ,\ i=0,\ \cdots,\ k \qquad (23)$$

Table 6 contains sample values of A(i,n,k) for k=4 and n=10,000.

It is now fairly easy to determine the total number of nodes T(n,k) in a B-tree of order 2k+1 containing n keys. We have

$$T(n,k) = \sum_{i=0}^{k} A(i,n,k)$$

$$= \frac{3\alpha}{1-\alpha}\ \ A(k)$$

$$= \frac{3(1-\alpha)\ (n-k)}{3+(2k-4)\ (1-\alpha)}$$

As an example $T(1000,4) = 182$.

B-TREES OF EVEN ORDER

We have so far assumed (as did Bayer and McCreight [1]) that the tree order m is odd. It is also possible to define a B-tree with an even order $m=2k$, $k\geq 2$. In this case, the nodes other than the root range in size from $k-1$ to $2k-1$. When a node splits, the two nodes created have sizes $k-1$ and k. The creation of nodes of unequal size has a significant effect on the expected relative number of nodes of each size, but most of the limiting values obtained for the odd order case are also valid for the even order case.

To begin the analysis, we must slightly modify the definition of $A(i,n,k)$, $i=0, \cdots, k$, to be the expected number of nodes of size $k-1+i$. Then Equations (2) and (3) are converted to

$$A(i,n+1) = A(i,n) + \frac{1}{n+1}\ ((k-1+i)\ A(i-1,n) - (k+i)\ A(i,n)),\ i=2, \cdots, k$$

$$A(0,n+1) = A(0,n) + \frac{1}{n+1}\ (2k\ A(k,n) - k\ A(0,n))$$

$$A(1,n+1) = A(1,n) + \frac{1}{n+1}\ (2k\ A(k,n) + k\ A(0,n) - (k+1)\ A(1,n))$$

Equations (4) and (5) are replaced by

$$\frac{A(i,n)}{A(j,n)} = \frac{(k-1+i)\ A(i-1,n) - (k+i)\ A(i,n)}{(k-1+j)\ A(j-1,n) - (k+j)\ A(j,n)},\ i, j=2, \cdots, k$$

$$\frac{A(0,n)}{A(j,n)} = \frac{2k\ A(k,n) - k\ A(0,n}{(k-1+j)\ A(j-1,n) - (k+j)\ A(j,n)},\ j=2, \cdots, k$$

$$\frac{A(1,n)}{A(j,n)} = \frac{2k\ A(k,n) + k\ A(0,n) - (k+1)\ A(1,n)}{(k-1+j)\ A(j-1,n) - (k+j)\ A(j,n)},\ j=2, \cdots, k$$

Further development follows in a fashion similar to that for the case of m odd, yielding the equations

$$\left(\frac{A(k-1)}{A(k)}\right)^{k+1} - 2\ \frac{A(k-1)}{A(k)} - 2 = 0 \tag{24}$$

$$A(i) = r^{k-i}(k)\ A(k),\ i=1, \cdots, k-1 \tag{25}$$

$$A(0) = 2\ A(k)/r(k) \tag{26}$$

where

$$r(k) = A(k-1)/A(k)$$

It is possible to approximate $r(k)$ numerically from (24), so that (25) and (26) provide formulas for $A(0), \cdots, A(k-1)$ as a function of $A(k)$. Moreover, it can be shown that

$$\lim_{k\to\infty} r^{k+1}(k) = 4,$$

so that $r(k) \approx 4^{\frac{1}{k+1}}$ for large k. Assuming this approximation to hold exactly, we get

$$A(i) = 4^{\frac{k-i}{k+1}}\ A(k),\ i=1, \cdots, k \tag{27}$$

$$A(0) = \frac{1}{2}\, 4^{\frac{k}{k+1}}\, A(k) \qquad (28)$$

Note that (27) is the same as (13) except that it does not include the case of i=0. Note also that (28) (i.e., the case i=0) is the same as (13) for i=0, except for the factor $\frac{1}{2}$.

Further derivations based on (27) and (28) can be made for m even, as was done for m odd using (13). We shall not present the derivations here, but will mention the most interesting results:

1. The expected number of splits during an insertion into the tree is approximately $\frac{1}{.75m}$, for both even and odd m.
2. The average storage utilization is slightly less than 70%, for both even and odd m.
3. The average tree height has the same formula for m even and m odd.
4. Table 7 contains values of A(i) for k=4 and m=8.

SIMULATION TESTS

Equations (13) and (23) are very basic to the entire analysis presented here. Accordingly, simulation tests were run to determine sample values for the averages predicted by (23). Parameters used were k=49 and n=20,000. One hundred trials were run, and the total number of nodes of each size was recorded. These empirical numbers were then compared with the corresponding theoretical values from (23) multiplied by 100. Figure 1 consists of a plot of the empirical and theoretical curves. As can be seen, the theoretical and empirical values are quite consistent.

SUMMARY

Several average performance measures for B-trees have been derived. These measures include the average storage utilization, average tree height, expected number of splits per insertion, expected total number of nodes, expected number of nodes of each size, and the expected number of accesses required for a retrieval or an insertion. We have shown that the average storage utilization is just under 70%, the average tree height is roughly logarithmic to the base .69m, and the expected number of splits per insertion is approximately $\frac{1}{.75m}$. We have presented simulation results to substantiate some of the theoretical analysis.

TABLE 1

Relative Values of A(i) for k=4

i	0	1	2	3	4
A(i)	3.0	2.3	1.7	1.3	1.0

TABLE 2

Values of p(i,4)

i	0	1	2	3	4
p(i,4)	.250	.227	.201	.174	.148

TABLE 3

Sample Values of t(k)

k	1	5	10	100	1000
t(k)	.429	.121	.0636	.00663	.000666

TABLE 4

Sample Values of v(k)

k	1	10	100	1000	∞
v(k)	.66667	.68893	.69345	.69394	.69401

TABLE 5

Sample Values of h(50,n)

n	1,000	10,000	100,000	1,000,000
h(50,n)	1.70	2.24	2.78	3.33

TABLE 6

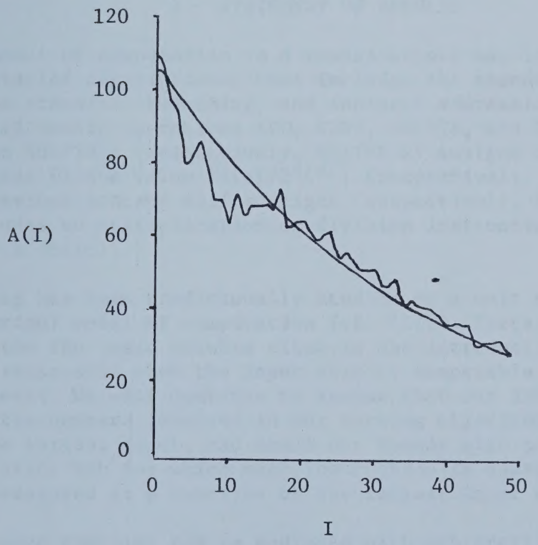
Absolute Values of A(i,1000,4)

i	0	1	2	3	4
A(i,10000,4)	591	448	339	257	195

TABLE 7

Relative Values of A(i) for m=8

i	0	1	2	3	4
A(i)	1.5	2.5	1.9	1.4	1.0



Comparison of Empirical and Theoretical Values of $A_{i,20000,49}$

Figure 1

REFERENCES

1. Bayer, R. and McCreight, E. Organization and maintenance of large ordered indexes. Acta Informatica 1(1972), 173-189.

2. Bayer, R. and Schkolnick, M. Concurrency of operations on B-trees. Acta Informatica 9(1977), 1-21.

3. Bayer, R. and Unterauer, K. Prefix B-trees. ACM Trans. on Database Systems 2(1977), 11-26.

4. Held, G. and Stonebraker, M. B-trees reexamined. Comm. ACM 21, 2(Feb. 1978), 139-143.

5. Knuth, D. The Art of Computer Programming, Vol. 3. Sorting and Searching. Addison-Wesley, Reading, MA, 1973, pp. 471-480.

6. Ullman, J. Principles of Database Systems. Computer Science Press, Potomac, MD, 1980, pp. 42-29.

7. Yao, A. On random 2-3 trees. Acta Informatica 9, 2(1978), 159-170.

AN UPPER BOUND FOR SORTING INTEGERS IN RESTRICTED RANGES

DAVID KIRKPATRICK
Department of Computer Science, University of British Columbia,
Vancouver, B.C., Canada V6T 1W5

ABSTRACT

Any set of n integers in the range [0,R-1] can be sorted in uniform time $O(n \lg \log_n R + n)$.[†] Our model of computation is a RAM with additions, subtractions and shifts, but no general multiplication or division instructions. The algorithm uses a technique of range reduction that may be viewed as a generalization of radix sorting.

1. INTRODUCTION

The general problem of sorting is one of the most fundamental (and well studied) problems in computer science. For the most general formulation of the problem it is natural to restrict one's attention to comparison-based algorithms. In this setting, it is well known that $O(n \lg n)$ operations (comparisons) are necessary and sufficient to sort an arbitrary n-tuple. It is also well known that if an n-tuple consists entirely of integers in the range [0,n-1] then it can be sorted in O(n) operations (exploiting the addressing capabilities of a random access machine). This is easily generalized to an O(nm) algorithm for sorting n-tuples of integers in the range $[0,n^m-1]$ [1]. It is natural to ask what is the largest range of integers for which a sorting algorithm can be constructed that runs asymptotically faster than the conventional $O(n \lg n)$ algorithms. We shall describe a new generalization of radix sorting that significantly extends the range of integers over which conventional sorting algorithms can be asymptotically improved.

2. STATEMENT OF RESULTS

Our model of computation is a random access machine (RAM) (see [1] or [2] for detailed descriptions) that includes the standard input/output, information transfer, branching, and indirect addressing capabilities, along with the arithmetic operations ADD, SUBT, SHIFTR, and SHIFTL. The shift instruction SHIFTR a (respectively, SHIFTL a) assigns to the accumulator (ie. register 0) the value $\lceil c(o)/2^{c(a)} \rceil$ (respectively, $c(0) \times 2^{c(a)}$), that is, its previous content shifted right (respectively, left) c(a) bits. Our model includes no multiplication or division instruction (other than that implied by a shift).

Sorting has been traditionally studied on a unit charged (ie. uniform cost criterion) model of computation (cf. [1]). Certainly this assumption underlies the the basic results cited in the introduction. Such a charging scheme is reasonable when the input size is comparable to the word size of real machines. We will continue to assume that our RAM is unit charged but note that the numbers involved in our sorting algorithms never exceed the size of the largest input, and hence our bounds also hold for a (possibly more realistic) RAM for which each instruction is charged in proportion to the size (measured as a function of the largest input value) of its operands.

We assume that our RAM is equipped with arbitrarily many registers each of which is initialized to 0. We will return to a discussion of this assumption in section 5.

[†] lg stands for $\log_2$ throughout this paper.

Let $S(n,R)$ denote the time required to sort an arbitrary set of n integers drawn from the range $[0,R-1]$. The basic result of this paper is the following:

Theorem 1. On a RAM, as described above, $S(n,R) = O(n \lg \log_n R + n)$.

This result should be contrasted with the straightforward multipass radix sorting algorithm (see [1]) which gives an $O(n \log_n R + n)$ upper bound on the same model. Our algorithm is based on a technique of systematic range reduction that is of interest in its own right.

3. RELATED WORK

In addition to the basic results mentioned in the introduction, there are a number of recent results that speak directly to the problem of sorting integers in restricted ranges. P. van Emde Boas [7] was the first to describe an efficient priority queue maintenance algorithm designed specifically to deal with integers from a restricted range. In subsequent papers [8,9] van Emde Boas' result has been refined to the following:

Theorem A. (van Emde Boas) There is a data structure supporting the full repertoire of priority queue operations over the range $[1,R-1]$ that was $O(R)$ space and $O(\lg \lg R)$ processing time per element processed. The space must be initialized in time $O(R)$ before the structure can be used.

It follows immediately from Theorem A that $S(n,R) = O(n \lg \lg R) + O(R)$, the second term being a one time preprocessing (initialization) charge. While this bound itself is not competitive with conventional techniques (as R increases the preprocessing cost dominates), it does point out that the incremental cost of sorting n integers in the range $[0,R-1]$ is only $O(n \lg\lg R)$.

Very recently S. Reisch [5] has shown that $S(n,2^{cn}) = O(c^{\frac{1}{2}}n)$ on a RAM that includes general multiplication and division operations. Reisch's algorithm differs from that of this paper not only in the underlying model of computation but also in its use of integers significantly larger than the inputs (thereby fully exploiting the uniform cost criterion). Reisch's arithmetic techniques complement the (essentially addressing) techniques of the present paper and it should not be surprising that these can be effectively combined toward an even stronger result on Reisch's more powerful model [6].

The upper bounds described above are complemented by lower bounds due to Paul and Simon [4] (and independently C. Rackhoff). In particular, they show that $S(n,n^n) = \Omega(n \lg n)$ on a RAM equipped with addition, subtraction, and multiplication but no integer division. Hence, for this model at least, we have a concrete upper bound on the range over which conventional sorting techniques can be asymptotically improved.

4. EFFICIENT RANGE REDUCTION

We know by conventional bucket sorting that $S(n,n) = O(n)$. To establish Theorem 1, it suffices to demonstrate the following "range-reduction" lemma:

Lemma 2. On a RAM as described in section 2, $S(n,2^k) \leq S(n,2^{\lceil k/2 \rceil}) + O(n)$.

Proof. Suppose we are given n integers in the range $[0,2^k-1]$.

Imagine that we have a RAM with $2^{\lfloor k/2 \rfloor} + O(n)$ registers each with a capacity of k bits (necessary and sufficient to represent numbers in the range $[0,2^k-1]$). Thus we can refer, without confusion, to the "first" i bits of a specific register. (Of course these bits can be accessed by appropriate SHIFTing).

The algorithm starts with $2^{\lfloor k/2 \rfloor}$ empty buckets. Input integers are assigned, in succession, to buckets; the bucket number is given by the integer's first $\lfloor k/2 \rfloor$ bits (that is, its first half). An integer is represented within its bucket by its second $\lceil k/2 \rceil$ bits (which together with the bucket number uniquely identify the original integer). Obviously, when k is large most buckets will remain empty, however each bucket must have the capacity for all n integers.

Suppose that some b buckets, we will refer to them as $B_1,B_2,\ldots,B_b$, are assigned one or more integers. (We call these "active" buckets). Let i_j be the index of B_j and let n_j be the number of elements assigned to B_j. The bucketing procedure builds a list structure of size $2^{\lfloor k/2 \rfloor} + O(n)$ (see Figure 1) containing:

i) a list Bucket [i], for $0 \leq i \leq 2^{\lfloor k/2 \rfloor}$ of the elements assigned to bucket number i; and

ii) a list Active-buckets of b active bucket records, each of which gives the bucket number and size (number of elements) of one active bucket.

Since the buckets are assumed to be initialized, it should be clear that the bucketing stage can be completed in constant time per element, that is O(n) steps in total.

The algorithm next sorts the list Active-buckets by size, that is, we can assume that $n_1 \leq n_2 \leq \ldots \leq n_b$. This can be done using a conventional bucket sort since $b \leq n$ and $n_i \leq n$, for all i. Thus the original problem has been reduced to the following:

i) sort the list Active-buckets on the bucket numbers $(i_1,i_2,\ldots,i_b)$;

ii) sort the list of elements associated with each active bucket; and

iii) combine the sorted bucket lists in the order specified by the sorted Active-bucket list to give the final sorted list.

Clearly, step iii) can be accomplished in O(n) steps. Note that all of the numbers to be sorted in steps i) and ii) lie in the range $[0,2^{\lceil k/2 \rceil}-1]$.

Let r denote the unique integer satisfying

$$\sum_{j=1}^{r} n_j < b \quad \text{and} \quad \sum_{j=1}^{r+1} n_j \geq b. \quad \text{Let } n_0 = b - \sum_{j=1}^{r} n_j$$

and $n_* = \max\{n_0,n_r\}$. A new bucket, call it B_0, is formed by removing n_0 elements from bucket B_{r+1}. The resulting sequence of buckets $B_0,\ldots,B_b$ has the properties:

(1) $1 \leq n_i \leq n_*$ for $0 \leq i \leq r$; and

(2) exactly n-b elements lie in the union of buckets $B_{r+1},\ldots,B_b$. The algorithm now,

(a) sorts bucket lists $B_0,B_1,\ldots,B_r$ individually, using any conventional ($O(n \lg n)$) sorting technique;

(b) coalesces bucket lists $B_{r+1},\ldots,B_b$ along with the list of active bucket numbers into one large list with exactly n elements. (Along with each element is recorded the bucket number from which it came.);

(c) sorts the coalesced list; and

(d) recovers the sorted individual lists from the result of step (c) (using the information recorded in step (b)).

Steps (b) and (d) are easily implemented in $O(n)$ steps. Step (c) is done recursively in $S(n,2^{\lceil k/2\rceil})$ steps. Hence, to complete the proof, it suffices to show that step (a) can be done in $O(n)$ steps. As described, step (a) takes

$$O\left(\sum_{i=0}^{r} n_i \lg n_i\right)$$

steps. We can assume that $n_* > 1$ (otherwise step (a) takes no time at all). By property (1) and straightforward maximization we have

$$\sum_{i=0}^{r} n_i \lg n_i \le \frac{b-r-1}{n_*-1}(n_* \lg n_*)$$

$$< (b-r)n_*.$$

But, since $n_j \ge n_*$ for $j > r + 1$,

$$n = \sum_{i=0}^{r+1} n_i + (b-r-1)n_*$$

$$> (b-r)n_*,$$

and hence step (a) requires at most $O(n)$ steps. □

It should be noted that the analysis of the above algorithm cannot be tightened. If k is even then in either of the extreme cases, that is when all of the integers are placed in the same bucket, or each is placed in its own bucket, the recurrence

$$S(n,2^k) = S(n,2^{\lceil k/2\rceil}) + O(n)$$

follows.

Our algorithm found it beneficial to "batch" together some of its sorting subproblems (see step (b)). It is interesting to note that while such batching is not advantageous for the general sorting problem, lower bounds like those of [4] suggest that, for integer sorting, this may be the most efficient approach to the solution of several (apparently independent) subproblems.

5. STORAGE CONSIDERATIONS

Careful inspection of the proof of lemma 2 reveals that it can (and perhaps should) be stated more precisely as follows:

Lemma 2'. On a RAM, as described in section 2, equipped with $2^{\lceil k/2 \rceil} + O(n)$ registers each of capacity k bits, $S(n,2^k) \leq S(n,2^{\lceil k/2 \rceil}) + O(n)$.

Obviously, the space requirements of the algorithm will be prohibitive in many situations. However, our scheme for range reduction can be generalized in a straightforward way to apply to less well endowed RAMS as well. Specifically,

Lemma 3. On a RAM, as described in section 2, equipped with $2^s + O(n)$ registers ($s \leq \lfloor k/2 \rfloor$) each of capacity k bits, $S(n,2^k) \leq S(n,2^{k-s}) + O(n)$.

This leads, in turn to the following generalization of Theorem 1.

Theorem 2. On a RAM, as described in section 2, equipped with $2^s + O(n)$ registers ($\lg n \leq s \leq \lfloor (\lg R)/2 \rfloor$) each of capacity $\lceil \lg R \rceil$ bits, $S(n,R) = O(n \lfloor \lg R \rfloor / s + n \lg(s/\lg n))$.

It is significant that our sorting algorithm has made effective use of a great deal more storage than time. It might be suspected that the initialization of memory would dominate the running time of our algorithm. While, in a sense, we are not obliged to defend our algorithm against this criticism (afterall, the standard RAM models [2] assume that memory has been initialized to zero, at no cost), it may be worth pointing out in this context what has become almost a "folklore" fact (that may or may not have influenced the original RAM specifications) namely:

FACT (Initialization of RAMs) - An uninitialized RAM (registers start with random values) can simulate an initialized RAM (all registers are initialized to zero) with only a constant factor of overhead.

This claim is, in essence, a restatement of problem 2.12 in [1]. The proof will, of course, be left to the reader.

6. CONCLUSIONS

We have shown that, given sufficient space, we can extend significantly the integer ranges over which conventional ($O(n \lg n)$) sorting algorithms can be asymptotically improved. This motivates the search for further improvement in both upper and lower bounds on the complexity of integer sorting. In particular, we are aware of no non trivial lower bounds for a model of computation that includes integer division (or, for that matter, just shifts).

Our results (especially in light of the discussion of section 3) lead naturally to the question - for which ranges of integers can conventional priority queue algorithms be asymptotically improved. The results of van Emde Boas et al [7,8,9] indicate that the question is non-trivial. Some preliminary results on this problem, along with a discussion of applications in computational geometry, are described in [3].

REFERENCES

1. Aho, A.V., Hopcroft, J.E., and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley, 1974.

2. Cook, S.A., and Reckhow, R.A., "Time bounded random access machines", J.C.S.S. 7,4 (1973), 354-375.

3. Keil, J.M., "Computational geometry on an integer grid", M.Sc. Thesis, Department of Computer Science, University of British Columbia, April 1980.

4. Paul, W.J., and Simon, J., "Decision trees and random access machines" Symposium uber Logik und Algorithmik, Zurich, 1980.

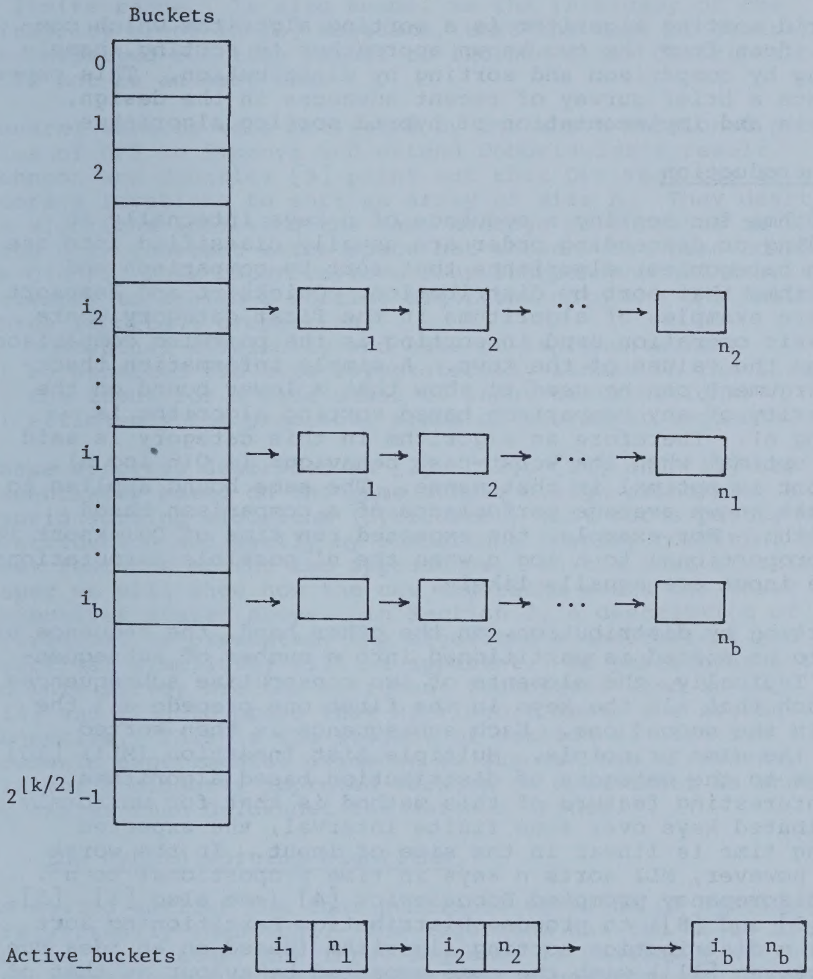
5. Reisch, S., "Efficient sorting by random access machines", preprint 1980, submitted for publication.

6. Reisch, S., and Kirkpatrick, D.G., "A note on integer sorting by random access machines", to be published.

7. van Emde Boas, P., "An O(n log log n) on-line algorithm for the insert-extract min problem", TR 74-221, Department of Computer Science, Cornell University, December 1974.

8. van Emde Boas, P., Kaas, R., and Zijlstra, E., "Design and implementation of an efficient priority queue", Math. Systems Theory 10(1977), 99-127.

9. van Emde Boas, P., "Preserving order in a forest in less than logarithmic time and linear space", Info. Processing Letters 6,4 (June 1977), 80-82.



Buckets
0
1
2
i_2
1
2
...
n_2
i_1
1
2
...
n_1
i_b
1
2
...
n_b
$2^{\lfloor k/2 \rfloor}-1$
Active_buckets
i_1
n_1
i_2
n_2
i_b
n_b

Figure 1

HYBRID SORTING ALGORITHMS

SELIM G. AKL AND HENK MEIJER
Queen's University, Kingston, Ontario, Canada

ABSTRACT

A hybrid sorting algorithm is a sorting algorithm which combines ideas from the two known approaches to sorting, namely sorting by comparison and sorting by distribution. This paper presents a brief survey of recent advances in the design, analysis and implementation of hybrid sorting algorithms.

1. Introduction

Algorithms for sorting a sequence of n keys internally in ascending or descending order are usually classified into one of two categories: algorithms that sort by comparison and algorithms that sort by distribution. Quicksort and Heapsort [10] are examples of algorithms in the first category where the basic operation used in sorting is the pairwise comparison between the values of the keys. A simple information theoretic argument can be used to show that a lower bound on the complexity of any comparison based sorting algorithm is $\Omega(n \log n)$. Therefore an algorithm in this category is said to be optimal when its worst-case behaviour is $O(n \log n)$. Heapsort is optimal in that sense. The same bound applies to the best known average performance of a comparison based algorithm. For example, the expected run time of Quicksort is also proportional to n log n when the n! possible permutations of the input are equally likely.

In sorting by distribution, on the other hand, the sequence of keys to be sorted is partitioned into a number of subsequences. Typically, the elements of two consecutive subsequences are such that all the keys in the first one precede all the keys in the second one. Each subsequence is then sorted using the same principle. Multiple List Insertion (MLI) [10] belongs to the category of distribution based algorithms. The interesting feature of this method is that for uniformly distributed keys over some finite interval, the expected sorting time is linear in the size of input. In the worst case, however, MLI sorts n keys in time proportional to n^2. This discrepancy prompted Dobosiewicz [4] (see also [1], [5], [6], [7] and [8]) to propose Distribution Partitioning Sort (DPS), a distribution sorting algorithm (based on an idea due to MacLaren [11]) with the same expected behaviour as that of MLI but whose worst-case run time is $O(n \log n)$. DPS starts by finding (in linear time) the smallest, largest, and median elements of the sequence to be sorted. Each of the intervals (min, median) and (median, max) is divided into n/2 intervals of equal length. The keys to be sorted are then distributed into the created intervals. The algorithm is now applied recursively to each of the intervals until the sequence is totally sorted.

Empirical results presented in [4] demonstrate that DPS is competitive in run time efficiency with the best known sorting algorithms. Due to its recursive nature and the complexity of the linear median finding algorithm, however, DPS suffers from a substantial amount of conceptual and practical overhead. Moreover its expected linear run time behaviour, as implemented, is restricted to sequences uniformly distributed over a finite range. It also seems, as the intricacy of the linearity proof in [4] suggests, that the task of generalizing the expected run time result to include other distributions will not be an easy one.

Several workers were motivated by the above mentioned properties of DPS to improve and extend Dobosiewicz's result. Johnson and Gonzalez [9] point out that DPS requires 2n extra storage locations to sort an array of size n. They describe an algorithm achieving the same average performance as DPS with only constant extra space but whose worst-case complexity is $O(n^2)$. This paper reviews four independent DPS-inspired attempts to design a sorting algorithm enjoying the following three important properties:
1) conceptual simplicity and ease of implementation;
2) an expected run time asymptotically linear in the size of the input for a wide class of input distributions;
3) efficiency for practical sizes of the input sequence.

These efforts, described in [2], [12], [14] and [15] are essentially based on the same simple design principle: a hybrid sorting algorithm (Hybridsort) that sorts partly by distribution and partly by comparison thereby combining the best features of the two approaches. In the rest of this paper we will show how the new algorithm meets the three objectives stated above. In Section 2, a description of Hybridsort is presented; the algorithm's theoretical and practical simplicity will be apparent. Of particular interest to this survey are the different techniques used in [2], [12], [14] and [15] to prove that Hybridsort meets the second objective. This is the subject of Section 3. Section 4 presents experimental evidence to support the efficiency claims: a slightly improved version of Hybridsort is shown to be faster than Quicksort for various inputs.

2. The hybrid sorting algorithm

As stated earlier the four published versions of Hybridsort are virtually identical. The algorithm has two major steps: in the first step, the input sequence is partially sorted by one iteration of a distribution sort algorithm; in the second step sorting is completed by comparison.

Let $\{x_1,x_2,...,x_n\}$ represent the sequence of real numbers to be sorted. We denote the minimum and maximum values of the input by u and v, respectively, and divide the interval (u,v) into n subintervals (buckets) of length (v-u)/n each. These buckets are numbered 1 to n.

Hybridsort Step 1: Distribute the elements of the input sequence into the n buckets; the bucket number for element x_i, where $u<x_i<v$, is obtained from

$$\text{bucket}\ (x_i) = \left\lceil \frac{x_i - u}{v-u} \cdot n \right\rceil .$$

(The ceiling function denoted by $\lceil x \rceil$ gives the smallest integer larger than or equal to the real x.)

Step 2: Sort each bucket using an optimal comparison based sorting algorithm. □

This algorithm is obviously very simple at the conceptual level. Experience also indicates that it presents no implementation problem either. Before turning to the expected run time analysis we briefly examine the worst-case complexity of Hybridsort.

Assuming that: 1) the computation of the ceiling function in Step 1 takes constant time, and that 2) every bucket containing t elements ($t>1$), is sorted in Step 2 using an algorithm whose run time is proportional to t log t, we have the following result.

Theorem 1: The worst-case complexity of Hybridsort is O(n log n). Proof: See [12]. □

3. Average-case analysis of Hybridsort

We start this section by stating the first "most natural" property of Hybridsort.

Theorem 2: If the numbers in the input sequence are independently drawn from a uniform distribution over the interval [0,1], then Hybridsort has a linear expected run time. Proof: See [12]. □

This expected behaviour of Hybridsort is reminiscent of that of DPS. In contrast with the complicated argument in [4], the linearity proof in [12] is surprisingly simple. In the following we review the various approaches taken in attempting to generalize the result in Theorem 2. Each approach essentially provides a characterization of those conditions under which Hybridsort has a linear average run time. The order of presentation adopted below is purely for ease of exposition.

3.1 Compact supports

In his complexity analysis, Devroye [2] assumes the input to be a sample $X_1, X_2, \ldots, X_n$ of independent identically distributed random variables with density function f. Moreover, he assumes that in Step 2 of Hybridsort a sorting method is used whose expected run time is given by the function g(t), where t is the number of elements in a bucket. This function must be convex and is allowed to grow faster than t but no faster than t^2. The results in [2] are based on the following definition.

Definition 1: A density function f is said to have compact support when there exists a finite K such that $\text{Prob}(|X|>K)=0$ where X has density f. □

Let C be a random variable representing the complexity of Hybridsort. The next theorem gives sufficient conditions for Hybridsort to exhibit a linear expect behaviour.

Theorem 3: If f has compact support (1)
and $\int g\,(f(x))\,dx < \infty$ (2)
then $E(C) = O(n)$.

Proof: See [2]. □

It is also shown in [2] that if f does not have compact support then E(C) must grow faster than n. This leads to the conclusion that the linear behaviour of Hybridsort is conditioned on f not having very high peaks or infinite tails. Typically, exponentially or normally distributed inputs will not be sorted in linear expected time. Devroye points out, however, that an order preserving mapping can sometimes be used to transform input numbers with density function f* into input numbers with density function f satisfying conditions (1) and (2). An example of this situation is given below.

Example: A density f* is said to have an exponentially dominated tail if there exist constants a, b, c>0 such that for all real numbers x,

$$f^*(x) \le a\,e^{-b|x|^c} .$$

Using the transformation

$$t(x) = \begin{cases} x/(1+x) & x\ge 0 \\ x/(1-x) & x<0 \end{cases}$$

it is possible to show that Hybridsort operating on $t(X_1)$, $t(X_2),\ldots, t(X_n)$, has a linear expect run time. □

An even stronger result is derived in [3]: when condition (1) is satisfied it is shown that condition (2) is not only sufficient but also necessary for the result in Theorem 3 to hold.

3.2 The Lipschitz Condition

We know from Theorem 2 that the average run time of Hybridsort is linear for inputs uniformly distributed over the interval [0,1]. This same behaviour can be obtained in the case where the input sequence, consisting of n independent identically distributed random variables $X_1,X_2,\ldots,X_n$, has another (known) density function f. All that is needed here is to apply the order preserving transformation

$$F(x) = \int_{-\infty}^{x} f(z)\,dz$$

and sort the sequence $F(X_1),F(X_2),\ldots,F(X_n)$, which clearly has a uniform distribution over the interval [0,1]. In most cases however, the distribution function F of the input sequence is not readily available, either because it cannot be computed or simply because f is not known.

One way to circumvent this difficulty is proposed by Weide

[15]. Let m be a positive constant independent of n, and let $X_{(1)}, X_{(2)}, \ldots, X_{(m)}$ be a sorted sample from the input sequence. This sample is randomly selected except for $X_{(1)}$ and $X_{(m)}$, which are to be taken as the minimum and maximum values, respectively, of the sequence $X_1, X_2, \ldots, X_n$. The function F can be estimated using

$$F_m(x) = \frac{1}{m-1}\left[i-1 + \frac{x-X_{(i)}}{X_{(i+1)}-X_{(i)}}\right] \quad \text{for } X_{(i)} \le x \le X_{(i+1)} \quad .$$

Now the sequence $F_m(X_1), F_m(X_2), \ldots, F_m(X_n)$ is sorted instead of the original sequence; this is equivalent to finding the bucket number of input element x_i from

$$\text{bucket } (x_i) = \lceil F_m(x_i) \cdot n \rceil \quad \text{for } X_{(i)} < x_i < X_{(m)} \quad .$$

Weide's treatment of Hybridsort is based on the assumption that each bucket is sorted in Step 2 using a polynomial time algorithm whose complexity is g(t) where t is the number of elements in the bucket. The only condition on g(t) is that it is bounded from above by a polynomial of degree at most ℓ; thus

$$g(t) \le \sum_{i=1}^{\ell} a_i t^i = \sum_{i=1}^{\ell} b_i \binom{t}{i} \quad .$$

The following definition is central to the analysis in [15].

Definition 2: A function F is said to satisfy a Lipschitz condition of order one with constant α if

$$|F(x) - F(y)| \le \alpha |x-y| \quad \text{for all real numbers x and y.} \ \square$$

Let C be defined as in Section 3.1. The next theorem gives sufficient conditions for Hybridsort to exhibit a linear expected behaviour.

Theorem 4: If f has compact support (3)
and F satisfies a Lipschitz condition of order one with constant α (4)
then E(C) = O(n).

Proof: See [15]. □

Although the results in [2] and [3] appear to be stronger, Weide's approach is in our opinion more appealing as it yields an efficient algorithm for most practical situations.

3.3 Uniform Boundness

Weide's observation was also made independently by the present writers [12]: If the input sequence $X_1, X_2, \ldots, X_n$ is drawn from a population with known density function f and if

$$F(x) = \int_{-\infty}^{x} f(z)\, dz$$

can be computed or approximated in constant time, then Hybridsort has a linear expected run time. The result derived in [12] is slightly more general than the ones in [2] and [15] and makes use of the following definition.

Definition 3: A function f of a real variable is said to be uniformly bounded if there exists a constant M such that

$$f(x) \leq M \text{ for all real numbers } x. \square$$

Let C have the same meaning as before and define

$$D = V-U$$

where $V = \max(X_1, X_2, \ldots, X_n)$

and $U = \min(X_1, X_2, \ldots, X_n)$.

The average behaviour of Hybridsort under very weak conditions on the input is described by the theorem below.

Theorem 5: If the elements of the input sequence are independently drawn from a population whose density function f is uniformly bounded, then the expected run time of Hybridsort is

$$O(n.h) \text{ where } h = E(D^{\varepsilon}) \text{ and } \varepsilon \in (0,1).$$

Proof: See [12]. □

It is easy to see that the following two results immediately follow from Theorem 5:

1) If f has compact support (i.e. D is bounded by a constant) then $E(C) = O(n)$ as in Theorem 3;

2) If f has exponentially vanishing tails, then $E(C) = O(n \log^{\varepsilon} n)$.

3.4 Integer Keys

As mentioned in the Introduction, a crucial step repeatedly applied in DPS is that of determining the median of a sequence of numbers. Van der Nat [14] observes that, for a sequence of length k, this operation is - on the average - more costly in computing time than merging two sorted sequences of length k/2 each using two-way merge [10]. He thus proposes the following hybrid sorting algorithm as an improvement on DPS.

Hsort Step 1: Split the input sequence arbitrarily into two subsequences of equal length. Step 2: For each of the two subsequences, find the maximum and the minimum elements. If they are equal, the subsequence is already sorted; otherwise do the following: a) Subdivide the interval between the maximum and minimum into a number of equally sized buckets; b) Distribute the elements of the subsequence into these buckets; c) Sort the contents of each bucket (if necessary) using Hsort recursively. Step 3: Merge the two sorted subsequences into one sorted sequence using two-way merge. □

Let C be the complexity of Hsort. For a sequence of length n, the worst case analysis of Hsort is very similar to that of DPS, yielding an upper bound of $O(n \log n)$ on C. Let us now assume that the number of buckets into which a subsequence of size k is divided is a linear function of k. The following theorem describes a particular situation for which the algorithm has a linear expected run time.

Theorem 6: If the elements in the input sequence are integers chosen from the interval $[1, \beta n]$ where β is a positive

constant, then $E(C) = O(n)$.
Proof: See [14]. □

Although Theorem 6 describes the behaviour of Hsort for a very special case, it should now be clear to the reader that the algorithm will have a linear expected run time for all distributions of the input possessing the properties given by definitions 1 and 3.

4. Practicality of Hybridsort

Deriving asymptotic results about the worst or average-case behaviour of an algorithm is usually only half of the task; implementing the algorithm in such a way that its performance matches the theory for reasonable sizes of the input is the other half. In [12], [14] and [15] various design decisions are discussed, and experiments are described whose results support those of the theoretical analyses.

An empirical study of Hybridsort's behaviour is presented in this section. What follows is a description of Hybridsort*, an efficient implementation of the algorithm in Section 2. Although attention is given here to some details, other aspects of the implementation, such as the choice of data structures, have been left out. Note that no assumption whatsoever is made about the distribution or range of the values in the input sequence.

Hybridsort* Let $\{x_1, x_2, \ldots, x_n\}$ denote the sequence of real numbers to be sorted.

Preprocessing Step (i) Find the minimum and maximum values of the input sequence; call these Q_0 and Q_4 respectively. (ii) Draw a random sample of size $\sqrt{n}$ from the input sequence (with Q_0 and Q_4 excluded). (iii) Find the first, second and third quartile values of the random sample; call these Q_1, Q_2, and Q_3 respectively. (iv) Define $F^*(x)$ as follows:

$$F^*(x) = \frac{1}{4}\left[i + \frac{(x-Q_1)}{(Q_{i+1}-Q_i)} \right] \quad \text{for } Q_i \le x \le Q_{i+1}$$

(i.e. $F^*(x)$ is a piecewise linear function approximating $F(x)$). If the slopes of two consecutive segments of $F^*(x)$ differ by less than

$$\frac{4}{Q_4-Q_0}$$

then replace these two segments by one.

Step 1: Distribute the elements of the input sequence into n buckets numbered $1,2,\ldots,n$. Element x_i is put in bucket j where

$$j = \lceil F^*(x_i) \cdot n \rceil \qquad \text{for } Q_0 < x_i < Q_4.$$

Step 2: Sort the contents of each bucket in nondecreasing order: If the bucket contains fewer than 10 elements
then use Insertion Sort [10];
otherwise use Median-of-3 Quicksort [10]. □

Hybridsort* was empirically compared to an extremely efficient version of Quicksort developed by Sedgewick [13]. Both algorithms were implemented in PASCAL and the tests were run on a Burroughs B6700 computer. The results of the various experiments are shown in Table 1. The table gives the average sorting time in seconds of both programs for different values of n and a particular distribution of input sequence.

Our results indicate that Hybridsort* is distinctly faster than Quicksort even for small values of n. The difference in performance between the two programs is more noticeable for files of large size (n>5,000). Thus we can only agree with the conclusion reached in [12], [14], and [15]: Because of its theoretical and practical properties the new hybrid sorting algorithm is highly recommended in all situations where speed of execution is an important factor.

References

[1] Burton, W., Comments on: Sorting by distributive partitioning, *Information Processing Letters*, Vol. 7, No. 4, June 1978, p. 205.

[2] Devroye, L., Average time behaviour of distributive sorting algorithms, Technical Report No. SOCS 79.4, School of Computer Science, McGill University, Montreal, Canada.

[3] Devroye, L. and Klincsek, T., Average time behaviour of distributive sorting algorithms, unpublished manuscript, 1979.

[4] Dobosiewicz, W., Sorting by distributive partitioning, *Information Processing Letters*, Vol. 7, No. 1, January 1978, pp. 1-6.

[5] Dobosiewicz, W., Author's reply to Warren Burton's comments on DPS, *Information Processing Letters*, Vol. 7, No. 4, June 1978, p. 206.

[6] Dobosiewicz, W., The practical significance of D.P. Sort revisited, *Information Processing Letters*, Vol. 8, No. 4, April 1979, pp. 170-172.

[7] Huits, M. and Kumar, V., The practical significance of distributive partitioning sort, *Information Processing Letters*, Vol. 8, No. 4, April 1979, pp. 168-169.

[8] Jackowski, L., Kubiak, R. and Sokolowski, S., Complexity of sorting by distributive partitioning, *Information Processing Letters*, Vol. 9, No. 2, August 1979, pp. 100.

[9] Johnson, D. B. and Gonzalez, T. F., Sorting numbers in linear expected time and constant extra space, Proceeding of the 16th annual Allerton Conference on Communications, Control and Computing, 1978, pp. 64-72.

[10] Knuth, D.E., *The Art of Computer Programming*, Vol. 3, Addison-Wesley, Reading, Massachusetts, 1973.

[11] MacLaren, M.D., Internal sorting by radix plus sifting, *Journal of the ACM*, Vol. 13, No. 3, July 1966, pp. 404 - 411.

[12] Meijer, H. and Akl, S.G., The design and analysis of a new hybrid sorting algorithm, Technical Report No. 79-79, Department of Computing and Information Science, Queen's University, Kingston, Canada.

[13] Sedgewick, R., Quicksort, Report STAN-CS-75-492, Stanford University, Stanford, California.

[14] Van der Nat, M., A fast sorting algorithm, a hybrid of distributive and merge sorting, Report No. 79-32, Institute of Applied Mathematics and Computer Science, University of Leiden, Leiden, The Netherlands.

[15] Weide, B.W., Statistical methods in algorithm design and analysis, Report CMU-CS-78-142, Department of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania.

n	Number of test runs	Average run time in seconds for inputs uniformly distributed over the interval [0,1]		Average run time in seconds for inputs normally distributed with zero mean and unit variance		Average run time in seconds for inputs exponentially distributed with parameter λ = 1	
		HYBRIDSORT*	QUICKSORT	HYBRIDSORT*	QUICKSORT	HYBRIDSORT*	QUICKSORT
100	100	0.065	0.049	0.072	0.048	0.066	0.050
250	100	0.158	0.146	0.176	0.142	0.162	0.148
500	100	0.314	0.326	0.352	0.316	0.331	0.332
1,000	80	0.634	0.746	0.702	0.702	0.652	0.736
2,500	60	1.534	2.086	1.721	2.003	1.644	2.092
5,000	40	3.004	4.655	3.381	4.655	3.314	4.600
10,000	20	6.196	10.632	6.805	10.748	6.222	10.013
15,000	10	8.752	15.732	10.533	16.795	8.927	16.273
20,000	4	11.537	21.492	14.992	21.596	11.938	22.475

TABLE 1

EXTERNAL SORTING REVISITED: APPLICATION OF DISTRIBUTIVE METHODS

DAVID J. HOGAN * AND BRUCE W. WEIDE
Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

Consider the problem of sorting a large number of large records according to some key field, so that the records end up in sorted order on disk. The conventional methods of solving this problem are merge-based. We explore the possibilities when particular partition-based strategies are employed that make use of distributional properties of the data. The major issues arise out of the desirability of overlapping essentially all computation and I/O.

1. INTRODUCTION

Suppose we have a single disk system on which it is required that we sort a large file of many large records (too many to fit in main memory at once). What is the best way to accomplish this? The traditional approach, adapted from a similar problem where the external storage medium is magnetic tape, is to use a merge-based method, in which "runs" of records in sorted order are formed and are merged to produce the final sorted file. There are many ways to produce these initial runs, many ways to merge them, and many variations that depend on the physical characteristics of the tapes and processor [3]. Here, we explore some issues that arise when an alternative approach using partition-based methods is attempted.

Many partition-based sorting algorithms exist [3]; we will concentrate on those requiring linear expected time when distributional information is known or can be inferred [1,2,4], and show how to adapt one such distributive method [4] to the external sorting problem. Our algorithm is very simple, and is described informally as:

```
procedure Sort (infile, outfile);
    file infile, outfile, subfile[ ];
    integer i, m;
    if infile cannot be sorted in main memory
        then begin
                Phase1 (infile, subfile, m);
                for i := 1 to m do Sort (subfile[i], outfile)
            end
        else Phase2 (infile, outfile)
end;
```

The purpose of Phase1 is to break the input file (infile) into enough smaller subfiles that each is extremely likely to be small enough to be completely sorted in main memory by what we have called Phase2. It must accomplish this subject to the constraint that every record in subfile[i]

This research supported in part by NSF MCS-79-12688.
* Also with Bell Laboratories, Columbus, Ohio.

is less than or equal to every record in subfile[i+1] in its key field, for every i, $1 \leq i < m$. The value of m is to be chosen by Phase1 so that each subfile is very likely to be small enough that the recursive call to Sort will directly invoke Phase2.

Phase2 is simply an internal sorting program that takes its input from infile and appends its output to outfile. This will work because no records of subfile[i+1] are appended to outfile until all records that come before them in sorted order have been output. This is the advantage of partition-based methods over merge-based ones; of course, in the latter, the partitioning step is particularly easy since the side constraint of order among the subfiles is unnecessary.

The purpose of this paper is to explore the reasons that a partition-based method might have advantages over standard merge-based methods if at least intermediate storage of files on disk is allowed. (Our major points are still applicable even if the original input file and the final output file are on tape.) These reasons stem from the relative ease with which computing and I/O can be overlapped if a distributive method is used. This simplicity is apparent both in the overall organization of the sort, where Phase1 is the distribution phase and Phase2 the internal sort within partitions, and in Phase2 itself.

Another promising aspect of the algorithm outlined above is that distributive methods provide ample opportunities for parallelism if multiple processors are available. Ordinary sorting algorithms have difficulty making full use of parallel processing either in the partitioning stage or the merging stage. With the distributive methods, merging is no problem (as is true for other partition-based methods), but neither is partitioning, so the full advantage of multiple processors becomes apparent. The issues that arise in the choice of an external sorting algorithm for a multiprocessor will not be extensively discussed here.

The next Section describes the model of computation and the nature of the input and output files, and the assumptions we make regarding both. The remaining Sections discuss the issues that are important in choosing implementations of Phase1 and Phase2.

2. MODEL OF COMPUTATION

Although the main intent of this paper is to explore issues rather than to present concrete results, we will define a precise model of computation that is assumed throughout. A single disk, big enough to hold the entire file to be sorted, is used as the sole I/O device, and all transfers of data to and from this disk take place as direct memory accesses managed by a file system. Input and output can be overlapped with computing with no perceptible effect on speed, but cannot be overlapped with each other.

The file manager keeps track of all information regarding the logical organization of files and their physical locations on the disk. It is capable of handling only sequential files, stored as linked lists of disk blocks, within which records of the file are stored contiguously. The file manager can be invoked to read in the next block of a file and to write out the next block (appending if the file already exists), and either operation takes time a + t*R, where R is the number of records transfered, and a and t are the average access time and transfer time for the disk, respectively. R will be equal to B, the maximum number of records in a block, except possibly for the last block of a file, which may not be full. There are no facilities for "scatter read" or "gather write"; rather, the information being transfered must also be contiguous in main memory.

The file being sorted consists of records having a key field (a real number uniformly distributed over some known range) plus a relatively large amount of additional extraneous information. We make absolutely no assumptions regarding the original order of the records; in particular, we do not assume that each input permutation is equally likely, only that the keys are identically uniformly distributed. (See [4], for instance, to determine how to transform virtually any set of keys so that this assumption is approximately satisfied. We will suppose that this conditioning step, if necessary, has already been performed for the file being sorted.)

The total number of records in the file is N and the total number of records for which there is main memory space is M. We will assume for simplicity that M is large enough so that the expected running time of our sorting algorithm is linear in N. It is, of course, possible to add a special algorithm that is invoked if this assumption is not satisfied, that simulates main memory using the disk and simply mimics any linear-expected-time sorting algorithm. This trick is possible because it takes only constant time to randomly access a disk. On the other hand, the efficiency of such a method would be terrible compared to that of our algorithm for any reasonable file and memory sizes, and it obscures the issues that distinguish external from internal sorting. For this reason, we simply assume that there is enough main memory to permit only a single invocation of Phase1 on the original file, and that the subfiles produced are each extremely likely to be small enough to be sorted internally by Phase2.

3. PHASE 1: DISTRIBUTION

Phase1 is intended as the distribution phase of a distributive sorting algorithm, in which the single large file is broken up into enough small files so that each is very likely to fit comfortably in main memory, where it can be sorted by the internal sorting algorithm in Phase2. This requires that m, the number of subfiles, be somewhat greater than N/M since statistical variation will lead to some subfiles being larger than N/m. Other factors are also involved in the choice of m, but this is the most important, since if m is too small, the Sort routine may have to be invoked recursively on several subfiles that turn out to be large, which would substantially increase the overall run time.

The main objective of Phase1 is achieved by pipelining the following operations: (1) Read records from infile into an input buffer; (2) Assign each record to one of the n subfiles by directly computing the index of that subfile as |key*m|+1 (assuming the keys are uniformly distributed between 0 and 1) and moving it to that subfile buffer; (3) Output records of each subfile. The pipelining is accomplished by reading into alternating input buffers so that computing need not wait for input under ordinary circumstances, and by having at least one output buffer for each of the m subfiles. (This is where the assumption about sufficient memory becomes important, since the choice of m to make each subfile sufficiently small may result in the output buffer space, which is at least m*B, exceeding M. Our assumption is that M is large enough that this does not occur.)

The computation time to assign a record to a subfile is minimal compared to the transfer time of a record, as is the time to physically move the record to the appropriate output buffer, so that it should be possible to overlap all computation with I/O and therefore make the total time of Phase1 approximately equal to the time to read in the entire file plus the time to write out all the subfiles. The only difficulty might be that computation could be delayed because a record was assigned to an output buffer that was currently being written out, so that no further input could occur because records were not being consumed from the input buffers. At the conclusion of the output of that subfile buffer, if no other such buffer could be output, no I/O could take place until either the input buffer were emptied or another output buffer filled. This is unlikely if the records arrive in random order and m is large, but might very well occur otherwise. As protection against this case, it seems worthwhile to double buffer each subfile output as well as input, virtually guaranteeing that I/O never waits for computation. This requires that M exceed at least 2*m*B for the algorithm to work as desired. Again, we note that alternative formulations and even more issues arise if this amount of main memory is not available.

One other interesting problem in Phase1 is the relative priorities that should be assigned to file input and output. If full output buffers are waiting and an input buffer is empty, it appears to be better in general to do input rather than output, especially with the double buffering scheme described above. The worst thing that could happen in this case is that computing could be delayed because there was no place to move a record, but then no further input could take place anyway and the output of some subfile buffer could begin. Clearly, the buffer to be written next should be the one holding up the computing, so that some flexible priority scheme among output buffers seems desirable. The advantage of this approach is that the system is still I/O bound. Since input and output cannot overlap each other, Phase1 must take essentially as much time as reading in the file plus writing out the subfiles, and our method approaches this bound with high probability.

4. PHASE2: INTERNAL SORTING

The job of Phase2 is to do an internal sort of the input file and append the records in sorted order to the output file. We accomplish this by reading the records into a large input buffer capable of holding the entire file, then moving records to one of (at least) two output buffers in sorted order, from which they are transfered back to disk, logically appended to the file consisting of all output of previous invocations of Phase2. All other internal record motion is avoided by using pointers to the records in the internal sorting algorithm.

This internal sort could presumably be accomplished by any of a number of standard algorithms, of which Quicksort is generally regarded as the best. However, recent developments in distribution-based methods indicate that such methods may be competitive with or, in some circumstances, superior to Quicksort. We will therefore consider such an algorithm as an alternative for the internal sort phase as well as being the basis for Phase1.

Remembering that the major difference between Phase2 and any ordinary internal sort is the I/O requirement, it is clear that any candidate for Phase2 should permit sorting to begin as soon as a small number of records have been read in, and should allow output of records in sorted order to start as soon as possible after the entire file has been read into memory. These requirements, of course, are due to the desirability of overlapping all computation and I/O, and in fact keeping the system I/O bound, since a clear lower bound on the time taken by Phase2 is the time required to read in plus the time to write out the entire file.

If Quicksort, for instance, is used here, there will be considerable difficulty in starting any computation before the entire file is in memory. We know of no published modification to Quicksort that allows partitioning to begin before the entire file is available. However, if a linked-list rather than array-based version of Quicksort is used, it is possible to start partitioning as the records are input, although the solution seems cumbersome. In essence, one begins with a single partitioning process that maintains two lists, one for all elements less than and the other for all elements greater than the partitioning element. As records are input, they are appended to these lists by the partitioning process. Whenever a list grows too long (a constant to be determined), a new partitioning process is spawned to further partition it. This new process receives its input records from the process that spawned it, and it operates recursively as its parent process does.

The linked lists are necessary because it is impossible to know before the fact how many records will fall into each partition, and because it is difficult to adjust pointers into an array if a fixed amount of space is allocated and then turns out to be insufficient. Of course, a simple array could be used if only the first partitioning stage of Quicksort were done as the records were read in, but then much of the subsequent partitioning would not be overlapped with I/O.

In short, an attempt to adapt Quicksort to the problem at hand faces considerable difficulty in the partitioning stage. Even if the rather inelegant first suggestion above is adopted, the number of levels of partitioning processes that a record would have to go through is difficult to predict, so that input might still be delayed by computing because of a poor choice of the maximum list size before a new process is spawned, or simply because of bad luck with the data. The essence of the problem is that total partitioning time is proportional to M log M on the average, while the total input time is proportional to M, so that it is not even necessarily possible to choose a small number as the maximum size of the subfiles that remain to be sorted, while overlapping all computing with input.

The other problem occurs with output. The partitions could be sorted in the order they occur in the sorted list, so that the first records of the sorted file could be moved to an output buffer and output while sorting of subsequent partitions took place. A balanced or I/O bound system here would dictate limits on the size of the lists being sorted at this point, and this choice might conflict with the similar limit imposed by the requirement during input and partitioning. For instance, an I/O bound input/partitioning stage might dictate that partitioning continue only until subfiles contained at most 50 records, while an I/O bound output stage might call for subfiles no larger than 10 records. Something has to give in this situation, and it is the completely overlapped computing requirement.

Our conclusion is that Quicksort is not ideally suited for Phase2, because it attempts to accomplish without making use of distributional information exactly what the distribution-based sorting algorithms do: to partition the file into many much smaller files that can be sorted by brute force and output in order. The main difference is that the distribution-based algorithms can do the partitioning in linear time, pipelining the input from disk with assignment to bins. This stage can be made I/O bound without restricting the number of bins (and therefore the expected size of the subfiles created), so that there will be no conflicting requirements with the output stage.

The input/distribution stage of Phase2 is simple. As records are read in, they are assigned to bins. In practice, the computation time per record required to do this is likely to be much less than the transfer time of a record, making this stage I/O bound, independent of the number of bins. Immediately following input of the last record, the first bin can be sorted by brute force, and its records moved in sorted order to an output buffer. Here, some balancing is required, so that the average rate at which records can be moved to the output buffers is at least equal to the transfer rate to disk. This is where the factors determining the choice of number of bins come into play. Assuming that physical parameters of the system allow the bins to be sufficiently small that this stage can also be kept I/O bound, the total time for Phase2 will be approximately equal to the time required to read in all the records, plus the time to write them out, plus the dead time for I/O, which is the time to sort enough bins by brute force to fill up an output buffer. This total is only very slightly more than the lower bound since the I/O dead time is extremely small compared to the transfer times. In fact, if computing is fast enough compared to I/O, the sorting within bins can be accomplished by insertion as the records are read in, making the I/O dead time before output begins essentially zero.

The decision about how to represent the bins (array of pointers or linked lists) presents another interesting choice. It is possible to reserve space only for as many pointers as there are records and rearrange them in place to form the bins, but this seems to require that a count of the number of records that will finally be assigned to each bin be known before rearrangement can take place. Such a solution, of course, defeats the goal of overlapping essentially all computation with I/O. Another array-based solution is to reserve space for an array of pointers substantially larger than the number of records. This approach seems promising, since it still permits pipelined input/distribution, although the steps that must be taken (albeit infrequently) to correct for an abnormally large number of records falling in some bin make the code rather long and difficult. The linked list solution also requires significantly more space for pointers than the number of records, but is very straightforward.

The second and third solutions have another nice feature: it is possible, and in fact easy, to make the sorting method a stable one. Records are placed in a bin in the order that they arrive, and a stable brute force method is used to sort within each bin. The same cannot be said for any known linear-expected-time method that uses only a constant amount of extra space. The price paid for this additional benefit, plus simplicity and totally overlapped distribution of records to bins, is that some space that otherwise could be used for storage of records is "wasted" on pointers. Fortunately, the larger the records, the less the importance of this drawback. It seems that some experimentation is necessary to resolve the trade-offs here.

5. CONCLUSIONS

We have seen that traditional methods do not seem as well suited to the external sorting problem as distributive methods, since the latter allow essentially all computation to be overlapped with I/O. This means that the system can be almost totally I/O bound, both during the distribution phase and during the internal sorting phase. Analysis and experimentation continue in an effort to determine how several trade-offs we have identified should be resolved in a real system.

REFERENCES

[1] Devroye, L. Average time behavior of distributed sorting algorithms. McGill Univ., School of Computer Science, Montreal, T.R. SOCS 79.4, Mar. 1979.

[2] Dobosiewicz, W. Sorting by distributive partitioning. Inf. Proc. Lett. 7, 1 (Jan. 1978), pp. 1-6.

[3] Knuth, D.E. The Art of Computer Programming, Vol. III: Sorting and Searching. Addison-Wesley, Reading, MA, 1973.

[4] Weide, B.W. Statistical Methods in Algorithm Design and Analysis. Carnegie-Mellon Univ., Dept. of Computer Science, Pittsburgh, PA, T.R. CMU-CS-78-142, Aug. 1978.

AN ON-CHIP SORTER/MERGER

D.T. LEE[1], H. CHANG[2] and C.K. WONG[2]

1. Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

2. IBM T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, New York 10598

ABSTRACT

Two generic record-permutation bubble devices - the bubble ladder and the bubble string comparator - have been reported in the literature. The former relies on the extensive use of external control lines while the latter relies solely on bubble-bubble interaction. This paper shows that equally efficient, but more versatile sorters can be constructed from the bubble string comparators without the control lines.

1. INTRODUCTION

Sorting requires the comparison of key values of records and the subsequent rearrangement of the records. Of all the permutation bubble devices depicted in the literature, the bubble ladder [2] is perhaps the most persistently investigated. Its structure consists of many fixed and equal length loops with a switch linking each pair of adjacent loops. By setting switches individually with external control lines, data can be rerouted among the loops. A number of algorithms have been developed for data management applications (FIFO, LIFO, automatic storage hierarchy, etc.) and sorting (odd-even [3,6], rebound [4], etc.). There are two shortcomings in the ladder. First, it does not provide the comparison function. Second, it derives its data steering capability from the extensive use of control lines.

Another recent development is the bubble string comparator [5]. The device compares two input data strings, sets a switch with memory, and guides the larger and smaller data strings respectively to designated max and min outputs. The device not only compares and steers data strings, but also performs these functions without external control lines. The reported work has not extended to sorting functions.

In this paper we shall pursue the following studies:

1. To construct a sorting device composed of bubble string comparators so as to eliminate the large number of control lines encountered in a bubble ladder.
2. To evolve efficient sorting algorithms for the new sorter, similar to those conceived for the ladder.

II. OPERATION OF THE SORTER

The basic component of the updown sorter to be described in this paper is the compare/steer unit shown in Figure 1. The unit consists of four interconnected cells. Cells A and B each receive two inputs, one from outside and another from an internal buffer cell. The two cells have no significant storage capacity. They are connected by a comparator, which immediately channels the comparison results to the other two cells, C and D. Buffer storages are available in C and D. They respectively store: C=min (A,B) and D=max (A,B), where A and B are the inputs to cells A and B respectively.

To accommodate the actions in the updown sorter, the compare/steer unit can function in downward or upward mode. See the solid and dotted arrows in Figure 1. In the downward (upward) mode, an outside key is loaded from the top (bottom) into cell A (B) of the unit, and an inside key previously at C (D) is loaded into B (A). After loading, the comparison is executed, and the minimum is delivered to C and the maximum to D, following the transition paths as indicated by the double arrows.

Loading and comparison are executed almost simultaneously. For convenience, we shall take the execution duration as a unit of time. All compare/steer units operate synchronously and take the same amount of time to operate. The notation $[t_i, t_{i+1})$ denotes the time interval from time instant t_i to time instant t_{i+1} but not including t_{i+1}.

Updown Sorter

The updown sorter is composed of a number of compare/steer units, one stacked on top of the other as shown in Figure 2. It can be shown that:

1. To sort n numbers, only $\lceil n/2 \rceil$ (i.e., the smallest integer no less than n/2) units (comparators) are needed.
2. The sorting time is completely absorbed in the input/output time. Moreover, we shall see in Section III that the output operation of a sorted sequence can be made to overlap with the input operation of the subsequent sequence, if the sequences are properly tagged.

Let us first explain how the updown sorter performs sorting in ascending order. In Figure 3, we show a complete histogram of the sorter during the sorting of an input of six numbers, 4, 3, 1, 6, 2 and 5 (4 leading).

Initially, at time t_0, the contents of the buffer cells in each compare/steer unit are all set to $+\infty$ or to any number larger than all possible input data. This initial state can be realized by continuously feeding $+\infty$ into the bottom unit before t_0. During the first cycle, the first number 4 is compared to $+\infty$ and routed to the upper right cell. During the second cycle, the number 3 is loaded and compared to 4; than the number 3 is routed to the upper right cell and 4 is routed to the lower left cell of the first unit. During the third cycle, as the third number is being loaded into the first unit, the number 4 is loaded into the second unit. In other words, the larger of the two numbers will be "pushed" out of the compare/steer unit in which it resides. At the end of time t_3, the upper right cell of the first unit contains 1 and the lower left cell contains 3; and the upper right cell of the second unit contains 4 and the lower left cell contains $+\infty$. In this fashion, the sorter continues to work. At the end of time t_5, all the six numbers have been loaded into the sorter, thus completing the input (downward) phase.

From time t_6 on, the output (upward) phase begins. Note that in the input phase the $+\infty$'s are pushed out of the bottom of the sorter; in the output phase the $+\infty$'s are pushed back into the sorter from the bottom. At the end of time t_6, the smallest number 1 is out and the second smallest 2 is in the upper right cell of the first unit awaiting to be output. In this output phase, the smaller of the two numbers within each compare/steer unit is "popped" up, leaving the unit it resides in and entering the unit on top of it. In the case of the top unit, the smaller number is delivered as output. Thus the sorter continues to put out the numbers in order. At the end of time t_{11}, all data in the sorter will have been output in ascending order as desired. At the same time, the sorter is automatically reset to its initial state (all $+\infty$) and is ready to accept the next input sequence.

To sum up, in input phase, the larger of the two numbers in each compare/steer unit is pushed down, and in output phase, the smaller of the two numbers is popped up. In general, the time interval $[t_0, t_n)$ corresponds to input phase and $[t_n, t_{2n})$ to output phase, where n is the number of keys to be sorted. While spending 2n units of time (or periods) to do input and output, we have automatically obtained a sorted sequence. The sorting is performed in a stack of n/2 compare/steer units.

III. ENHANCEMENTS - REVERSING, OVERLAPPING AND MERGING

In this section we shall describe methods of enhancing the operation described in the previous section.

Descending-Order Sorting

In the previous section we described sorting in ascending order. To sort in descending order, we shall exchange the input and output ports. That is, let the input data enter the sorter from the lower right end (i.e., where the number $+\infty$ enters) and output data come out from the lower left end (i.e., where $+\infty$ comes out). In addition to the I/O port exchange, the sorter must be initialized to contain a number known to be smaller than the input data.

Overlapping Consecutive Sorting Sequences

Now we shall overlap the input of the next sequence with the output of the current sequence. First, the next sequence is to be sorted in the opposite order; namely, if the current output is in ascending order, the next output will be in descending order and vice versa. Second, the data entered from the top are to be prefixed with a tag bit 0 and the data entered from the bottom prefixed with a 1. As the comparator compares the tag bits together with the keys themselves, those keys with tag bit 1 will be judged larger than those with tag bit 0. Thus the two sequences will be kept distinct before, during and after sorting.

Merging

What if we have a sequence of numbers whose size (m) is larger than the updown sorter (size n) can handle? A straightforward method is to divide the sequence into k ($k=\lceil m/n \rceil$) subsequences, each of size n (with the possible exception of the last one), and feed them into the sorter. After we have obtained k sorted subsequences, we shall use the sorter to do a k-way merge if $k \leq n$. If $k > n$, then we perform n-way merge as many passes as necessary. (See Knuth [1].) With no loss of generality let us assume $k \leq n$.

We have two cases to consider. First, suppose that we have buffer space large enough to hold the merged output. In this case the standard k-way merge technique applies. That is, we feed the k smallest numbers, one from each subsequence, into the sorter. Output the smallest one. Then feed in the next number from the subsequence to which the previous output number belongs. The detection of the originating subsequence is by tag which is affixed as the least significant bits on each number. Obviously, the total time for sorting these m numbers is $(k+1)n + 2m$ where $k=\lceil m/n \rceil$, and the first term is for sorting k subsequences and the second term for merging.

Second, suppose that we do not have extra storage in addition to the space for the input data. In this case we need to do an "in-place" merging. Imagine that we have k data chips (numbered 1 through k). It is desired to arrange the output sequence after sorting such that the first chip holds the n smallest numbers and the second chip the next n smallest numbers, and so on.

We shall adapt the idea of odd-even merge used in Batcher's sorting network [1] to perform the in-place merging. First let us describe the odd-even merge:

1. Given two sorted sequences in ascending order $a_1, a_2, \ldots, a_{2p}$ and $b_1, b_2, \ldots, b_{2p}$, $p \geq 1$.
2. Merge odd sequences $a_1, a_3, \ldots, a_{2p-1}$ and $b_1, b_3, \ldots, b_{2p-1}$ to form a sorted sequence $v_1, v_2, \ldots, v_{2p}$. Similarly, merge even sequences $a_2, a_4, \ldots, a_{2p}$ and $b_2, b_4, \ldots, b_{2p}$ to form a sorted sequence $w_1, w_2, \ldots, w_{2p}$. Note that v_1 and w_{2p} are respectively the smallest and largest numbers.
3. Compare w_i and v_{i+1} for $i = 1, 2, \ldots, 2p-1$. Let $c_i = \min(w_i, v_{i+1})$ and $d_i = \max(w_i, v_{i+1})$. Then the sequence $v_1, c_1, d_1, c_2, d_2, \ldots, d_{2p-1}, w_{2p}$ is the final sorted sequence resulting from the merging.

The odd-even merge algorithm can be generalized to merge two sequences where elements a_i and b_i are replaced by two sorted subsequences respectively [1]. The algorithm to merge sequences follows:

1. Given two ascending order sorted sequences each composed of subsequences with the same number of keys, $A_1, A_2, \ldots, A_{2p}$ and $B_1, B_2, \ldots, B_{2p}$, $p \geq 1$.
2. Merge odd sequences $A_1, A_3, \ldots, A_{2p-1}$ and $B_1, B_3, \ldots, B_{2p-1}$ to form a sorted sequence $V_1, V_2, \ldots, V_{2p}$. Similarly, merge even sequences $A_2, A_4, \ldots, A_{2p}$ and $B_2, B_4, \ldots, B_{2p}$ to form a sorted sequence $W_1, W_2, \ldots, W_{2p}$.
3. Instead of comparison, merge W_i and V_{i+1} for $i = 1, 2, \ldots, 2p - 1$. The smaller and larger halves of the merged sequence are respectively denoted by C_i and D_i. Then the sequence

   $$V_1, C_1, D_1, C_2, D_2, \ldots, C_{2p-1}, D_{2p-1}, W_{2p}$$

   is the final sorted sequence resulting from the merging.

Now we are ready to spply the generalized odd-even merge. Consider the case of two data chips, each with a storage capacity equal to that of the sorter, and each storing a sorted sequence of n numbers. We divide A into two halves A_1 and A_2 such that A_1 has the smaller n/2 numbers and A_2 has the larger. Similarly B is divided into B_1 and B_2. The data movement to execute the generalized odd-even merge is illustrated by the table below. Initially, the sorter is assumed to have n 0's, each with a tag bit 1, as indicated in the last column in which each bit 0 or 1 stands for n/2 tag bits of 0's or 1's. Steps 1 and 2 feed A_1 and B_1 into the sorter from the top, with each number being tagged with a 0. In step 2, i.e., at the end of the input of B_1, the sorter is ready to begin delivery of the sequence V_1V_2 (still being sorted). In step 3, while the sorter is delivering V_1 to Chip 1 from the top, the sequence B_2 of Chip 2 is read into the sorter from the bottom, with each number being tagged with a 1. Step 4 feeds A_2 into the sorter and obtains V_2 from the sorter. In step 5, W_2 is output to Chip 2, but only 0's are fed into the sorter. Now, V_2 can be fed into the sorter from the bottom, with each number being tagged with a 1. Notice that the sequence V_2 is fed into the sorter from the bottom. The reason is that V_2 is to be merged with W_1 and W_1 are tagged with 1. Steps 7 and 8 output D_1 and C_1 into Chip 2 and Chip 1 respectively, thereby completing the merge process. Note that the sorter is in a different "state" in which the tag bits are 0's instead of 1's.

STEP	CHIP 1	CHIP 2	SORTER	TAG BITS
0	A_1 A_2	B_1 B_2	00(initially)	11
1	A_2	B_1 B_2	A_1 0	01
2	A_2	B_2	B_1 $A_1 \rightarrow V_1$ V_2	00
3	A_2 V_1	-	V_2 B_2	01
4	V_1	V_2	B_2 $A_2 \rightarrow W_1$ W_2	11
5	V_1	V_2 W_2	0 W_1	01
6	V_1	W_2	W_1 $V_2 \rightarrow C_1$ D_1	11
7	V_1	D_1 W_2	0 C_1	01
8	$V_1 C_1$	D_1 W_2	00	00

As can be seen from the table, each step is an I/0 operation involving n/2 numbers. Thus, the total time for merging 2n numbers is $8 * n/2 = 4n$ periods. The performance of the scheme would become worse when the number of sorted sequences to be merged becomes larger.

In general, suppose we have $k = 2^p$ chips of data to be merged, where p is an integer greater than 1. We can use a divide-and-conquer technique to merge the first k/2 chips and the second k/2 chips of data recursively to obtain two sorted sequences A and B, each with $n * k/2$ numbers and then use the generalized odd-even merge algorithm to merge these 2 sequences by dividing them into k subsequences each, i.e., A_1, A_2, ..., A_k and B_1, B_2, ..., B_k such that each subsequence has n/2 numbers and performing the odd-even merge operations. If T(k) denotes the time required to merge k data chips then we have the following recurrence relation,

$$T(k) = 2*T(k/2) + M(k, k)$$

$$T(2) = 8$$

where M (k, k) denotes the time to merge A_1, A_2, ..., A_k and B_1, B_2, ..., B_k. Since M (k, k) can be expressed as

$$M(k, k) = 2*M(k/2, k/2) + 2*k$$

$$M(4, 4) = 24$$

We can solve for M (k, k) and T (k). In fact, $M(k, k) = 2(\log_2 k + 1)k$ and $T(k) = (\log_2 k)(\log_2 k + 3)k$. Therefore, the time for merging k chips of data is $(\log_2 k) * (\log_2 k + 3) * k * n/2$ periods.

IV. DISCUSSIONS

As a summary, the updown sorter is compared to the rebound sorter:

	UPDOWN SORTER	REBOUND SORTER
Hardware	Bubble string comparator (compare/steer)	Bubble ladder (steer)
	Bubble switch (internal control) $\lceil n/2 \rceil$ sets	Conductor switch (external control) $(n - 1)$ sets
Algorithm	Every comparator active during every cycle	Every comparator active every other cycle
	Sorting within input/output time	Sorting within input/output time
	Bi-directional	Unidirectional
	Overlapping sequences	
Nature of data	Serial input/output	Serial input/output
	Fixed-length key or key record	Fixed-length record

The most important hardware difference is the number of control conductors required by the two sorters.

REFERENCES

1. D. E. Knuth, "The Art of Computer Programming," Vol. 3, "Sorting and Searching," Addison-Wesley, Reading, Mass., 1973.
2. C. Tung, T. C. Chen, and H. Chang, "Bubble Ladder for Information Processing," *IEEE Trans. on Magnetics*, Vol. II, No. 5, pp. 1163-1165, Sept. 1975.
3. T. C. Chen, K. P. Eswaran, V. Y. Lum, and C. Tung, "Simplified Odd-Even Sort Using Multiple Shift-Register Loops," *Int'l J. Comp. and Info. Sci.*, Vol. 7, No. 3, pp. 295-314, Sept. 1978.
4. T. C. Chen, V. Y. Lum, and C. Tung, "The Rebound Sorter: An Efficient Sort Engine for Large Files," *Proc. 4th International Conference on Very Large Data Bases*, pp. 312-318, 1978.
5. B. Antonini, M. Bacchi, and G. Bongiovanni, "A Bubble String Comparator for Information Processing," *IEEE Trans. on Magnetics*, Vol. 15, No. 3, pp. 1183-1185, July 1979.
6. F. Y. Chin and K. S. Fok, "Fast Sorting Algorithms in Uniform Ladder (Multiple Shift-Register Loops)," *IEEE Trans. on Computers*, Vo. C-29, No. 7, pp. 618-631, July 1980.

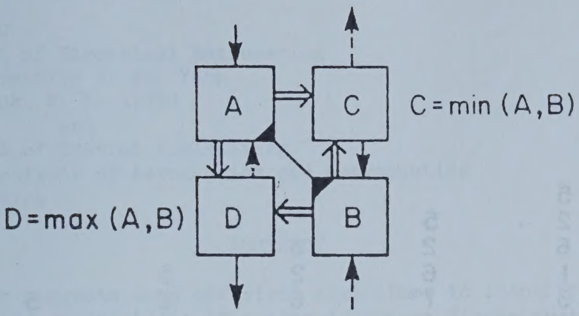

A
C
C = min (A,B)
D = max (A,B)
D
B

Fig. 1

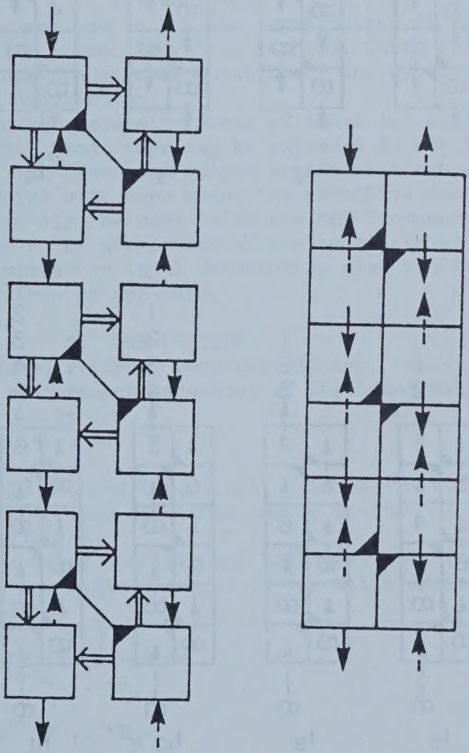


Fig. 2

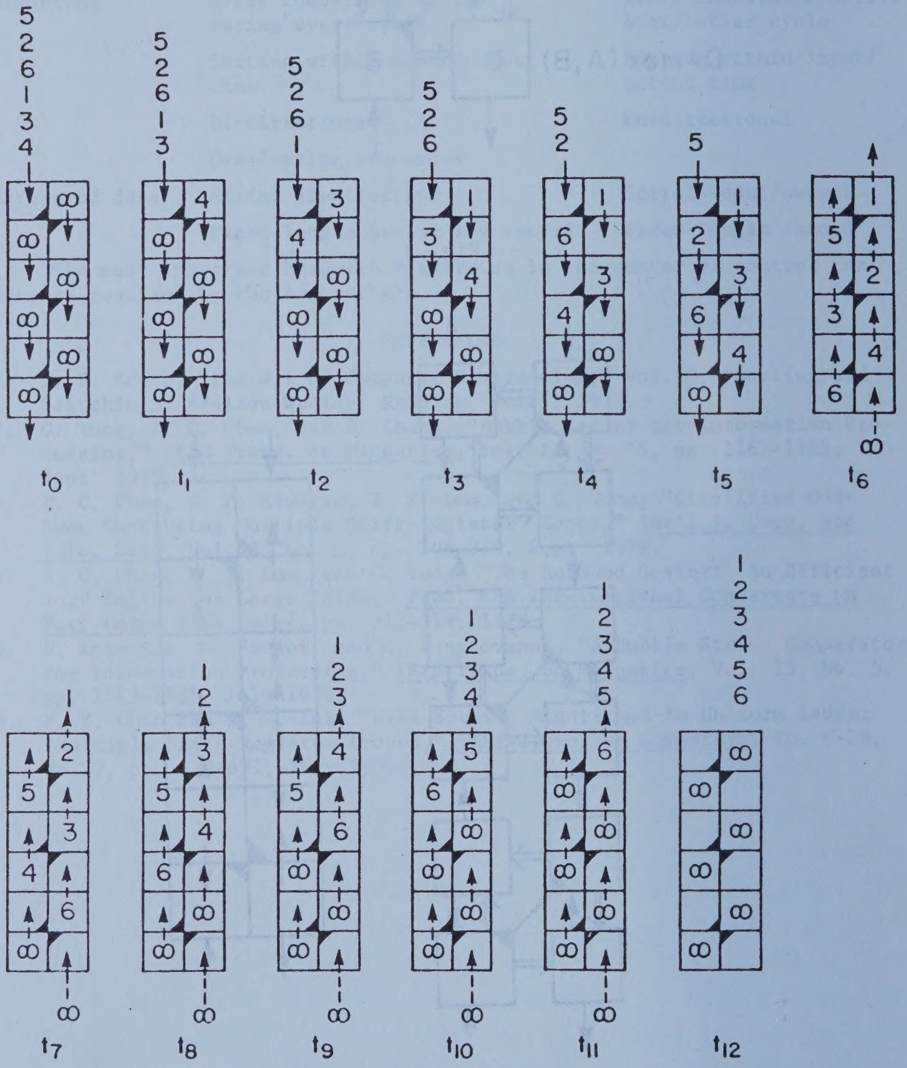

5 2 6 1 3 4
t_0
t_1
t_2
t_3
t_4
t_5
t_6
t_7
t_8
t_9
t_{10}
t_{11}
1 2 3 4 5 6
t_{12}
∞

Fig. 3

IDENTIFICATION OF MULTIVARIABLE LINEAR SYSTEMS

JIWEN WANG
Department of Electrical Engineering
State University of New York
Stony Brook, N. Y. 11794
and
Department of Control Engineering
Peking Institute of Aeronautics and Astronautics
Peking, China

ABSTRACT

This paper suggests some efficient algorithms to identify the input-output difference descriptions of a time-invariant linear system from its input-output measurements. To accomplish both structural identification and parameteric identification in one step and to use row-operation of matrix instead of inverse of matrix are the two features of the algorithms suggested. They are applicable to either noise or noise-free cases.

1. INTRODUCTION

This paper will discuss how to get the input-output difference description of a system from its normal operating records. There are two algorithms introduced. The state-space representation of the system can be obtained[2]-[4],[8] from such description.

To save computation the recursive forms of those two algorithms will be suggested. On the other hand, they can be extended to use least-squares estimator. Therefore if the input and output sequences are corrupted by some additive uncorrelated noise with zero mean, the extensive ones can be applied. In addition, they can also be used to obtain the "compensated" least-squares estimator[2] while the statistics of the noise are known.

The conditions required for input sequence is also studied and some numerical examples are given as appendix.

2. THE PROBLEM

Consider a noise-free, r-input, p-output, linear, time-invariant, discrete-time system. The measurement sequences of input and output vectors from normal operating records are

$$u(k),\ u(k+1),\ u(k+2),\ \ldots. \qquad (2\text{-}1)$$

$$y(k),\ y(k+1),\ y(k+2),\ \ldots. \qquad (2\text{-}2)$$

where $u(k+m) \in R^r$, $y(k+m) \in R^p$, $m=0,1,2,\ldots,N$, k is the initial time.

The input-output difference equations can be written as

$$A(z)y(k+m) = B(z)u(k+m) \qquad (2\text{-}3)$$

where z^{-1} is the unitary delay operator and both A(z) and B(z) are polynomial matrices in z. Let the highest degree of the all elements in the i-th row of A(z) be w_i and define

$$w \triangleq \max\{w_1, w_2, \ldots, w_p\}. \qquad (2\text{-}4)$$

Then A(z) and B(z) can be expressed as

$$A(z) = A_0 z^w + A_1 z^{w-1} + \ldots + A_w \qquad (2\text{-}5)$$

and

$$B(z) = B_0 z^w + B_1 z^{w-1} + \ldots + B_w. \qquad (2\text{-}6)$$

(3-3) can be rewritten as

$$-B(z)u(k+m) + A(z)y(k+m) = 0. \qquad (2\text{-}7)$$

Let $m=0,1,2,\ldots(N-w)$, then $(N-w+1)$ equations can be obtained and written as

$$\left[-B_w\ A_w\ -B_{w-1}\ A_{w-1}\ \cdots\ -B_0\ A_0\right]\cdot\begin{bmatrix} u(k) & u(k+1) & \cdots\cdots & u(k+N-w) \\ y(k) & y(k+1) & \cdots\cdots & y(k+N-w) \\ u(k+1) & u(k+2) & \cdots\cdots & . \\ y(k+1) & y(k+2) & \cdots\cdots & . \\ . & . & \cdots\cdots & . \\ . & . & \cdots\cdots & . \\ u(k+w) & u(k+w+1) & \cdots & . \\ y(k+w) & y(k+w+1) & \cdots & . \end{bmatrix}$$

$$= 0_{p\ x\ (N-w+1)}\cdot \tag{2-8}$$

Define p x (p+r)(w+1) matrix

$$C \triangleq \left[-B_w\ A_w\ -B_{w-1}\ A_{w-1}\ \cdots\ -B_0\ A_0\right] \tag{2-9}$$

and 1 x (N-w+1) block-rows

$$U(k+m) \triangleq \left[u(k+m)\ u(k+m+1)\ \ldots\ u(k+N+m-w)\right] \tag{2-10}$$

$$Y(k+m) \triangleq \left[y(k+m)\ y(k+m+1)\ \ldots\ y(k+N+m-w)\right] \quad m = 0,1,\ldots w$$

and 1 x (N-w+1) rows

$$U_i(k+m) \triangleq \left[u_i(k+m)\ u_i(k+m+1)\ \ldots\ u_i(k+N+m-w)\right], i = 1,..,r \tag{2-11}$$

$$Y_i(k+m) \triangleq \left[y_i(k+m)\ y_i(k+m+1)\ \ldots\ y_i(k+N+m-w)\right], i = 1,..,p \tag{2-12}$$

and (p+r)(w+1) x (N-w+1) matrix

$$E_w^0 \triangleq \begin{bmatrix} U(k) \\ Y(k) \\ . \\ . \\ U(k+w) \\ Y(k+w) \end{bmatrix}. \tag{2-13}$$

Then (2-8) can be expressed as

$$C\cdot E_w^0 = 0_{p\ x\ (N-w+1)}\cdot \tag{2-14}$$

According to [2], the i-th row's diagonal element of A_{wi} always is unit and all elements on its right hand side in the i-th row of C are zero. Therefore it follows that that the (2-8) or (2-14) holds means the $Y_i(k+w_i)$ is dependent on all its previous rows.

The problem is how to determine the set of indexes $w_1,\ldots,w_p$ and the non-zero elements of C uniquely when the input/output measurement sequences (2-1) and (2-2) are given. Usually they are refered as the structural identification and parameteric one respectively. Existing methods are often devided two steps to perform them. In this paper it will be accomplished in one step by row-operation of matrix.

3. BASIC ALGORITHMS

(1) Algorithm 1

At first w^*, the estimate value of w, is assumed since the actual value of w is not known. Then the $E_{w^*}^0$ is a $(p+r)(w^*+1)$ x $(N-w^*+1)$ matrix. Define a new matrix F^0 by setting a compatible identity matrix D^0 on the right hand side of $E_{w^*}^0$ as

$$F^0 \triangleq \left[E_{w^*}^0 \,\vdots\, D^0\right]. \tag{3-1}$$

Let e_{1g}^0 be the maximal element in absolute value of the first row of $E_{w^*}^0$. Then add the whole first row of F^0 multiplied by $(-e_{tg}^0/e_{1g}^0)$, t=2,3,...., $(p+r)(w^*+1)$, to the t-th row respectively. In this way, the g-th column, except the first element, of $E_{w^*}^0$ become a zero column and the first column, except the first element, of D^0 is turned into non-zero column. After the first step of row-operation the new matrix is denoted by

$$F^1 = \left[E_{w^*}^1 \,\vdots\, D^1\right]. \tag{3-2}$$

The D^1 has just recorded the first step of row-operation because the t-th element of its first column is exactly $(-e_{tg}^0/e_{1g}^0)$ as

$$D^1 = \begin{bmatrix} 1 & & & \\ d_{21}^1 & 1 & & 0 \\ \vdots & & \ddots & \\ d_{t1}^1 & 0..0 & 1 & \\ \vdots & & & \ddots \end{bmatrix} \tag{3-3}$$

where $d_{t1}^{1} = -e_{tg}^{0}/e_{1g}^{0}$. Let e_{2j}^{1} be the maximal element of the second row of E_{w*}^{1} in absolute value. Then add the whole second row of F^1 multiplied by $(-e_{tj}^{1}/e_{2j}^{1})$, $t = 3,4,\ldots,(p+r)(w^*+1)$, to the t-th row respectively. Similarly , the j-th column, except the first two elements, of E_{w*}^{1} becomes a zero column and the second column, except the first two elements, of D^1 is turned into non-zero column. After the second step of row-operation, the new matrix is denoted by

$$F^2 = \left[E_{w*}^{2} \,\vdots\, D^2 \right] . \qquad (3\text{-}4)$$

The D^2 has just recorded the first two steps of row-operation as follows

$$D^2 = \begin{bmatrix} 1 & & & & & \\ x & 1 & & & & 0 \\ x & x & 1 & & & \\ \cdot & \cdot & & \ddots & & \\ d_{t1}^{2} & d_{t2}^{2} & 0 & \ldots & 0 & 1 \end{bmatrix} \qquad (3\text{-}5)$$

where $d_{t1}^{2} = (-e_{tg}^{0}/e_{1g}^{0})+(e_{tj}^{1}/e_{2j}^{1})(e_{2g}^{0}/e_{1g}^{0})$, $d_{t2}^{2} = (-e_{tj}^{1}/e_{2j}^{1})$. It indicates that the t-th row of E_{w*}^{2} consists of the first row multiplied by d_{t1}^{2}, the second row multiplied by d_{t2}^{2} and the t-th row of E_{w*}^{0}.

In this process, after m steps of row-operation D^m and E_{w*}^{m} are obtained and the each row of D^m indicates how is the correspondent row of E_{w*}^{m} consisted from the rows of E_{w*}^{0}.

If there is no non-zero element in the (m+1)-th row of E_{w*}^{m}, a dependent row has been found and the (m+1)-th row of D^m indicates the coefficients of linear combination from the first m rows of E_{w*}^{0}. In this case, let

$$D^{m+1} = D^m$$
$$E_{w*}^{m+1} = E_{w*}^{m} . \qquad (3\text{-}6)$$

Then proceed to the next step of row-operation. The (m+1)-th column, except the first (m+1) elements, of D^{m+1} is still a zero column.

If the input sequence is persistently exciting, which will be examined in detail in the next section, the rows of $U(k+m)$ $(m = 0,1,\ldots,w^*)$ can not be turned to zeros after such row-operation. Therefore while the row-operation are finished, $D^{(p+r)(w^*+1)-1}$ and $E_{w*}^{(p+r)(w^*+1)-1}$ can be devided to $2w^*$ blocks just like E_{w*}^{0}(see(2-13)), but only those blocks which correspond to $Y(k)$, $Y(k+1),\ldots,Y(k+w^*)$ are interested and numbered with $0,1,\ldots,w^*$. Then check the each numbered block in turn. For example, the i_1-th row of the block 2 is found to be dependent one. It means that $w_{i_1} = 2$. Of course, all i_1-th rows of succedent numbered blocks also are dependent. Then the i_2-th and i_3-th row of block 3 are found to be new dependent ones, therefore $w_{i_2}=3$, $w_{i_3}=3$. This process is not finished until that $w_{i_p}= w$ is found. In actual computation, the computer should be told that when w_{i_1} is determined, $Y_{i_1}(k+w_{i_1}+1)$, $Y(k+w_{i_1}+2),\ldots$ should be excluded in the succedent steps of row-operation, when w_{i_2} is determined, $Y_{i_2}(k+w_{i_2}+1)$,$Y_{i_2}(k+w_{i_2}+2),\ldots$ should be done,...., when $w_{i_p}= w$ is determined, the whole row-operation should be finished.

$D^{(p+r)(w^*+1)-1}$ also plays a role of recorder like D^1, D^2,... If w_{i_1}, $w_{i_2},\ldots,w_{i_p}$ are found, the $((p+r)w_{i_1}+r+i_1)$-th row of $D^{(p+r)(w^*+1)-1}$ is just the i_1-th row of C,.... the$((p+r)w_{i_p}+r+i_p)$-th row of $D^{(p+r)(w^*+1)-1}$ is just the i_p-th row of C. In this way, C has completely been determined.

(2) Algorithm 2

It is often for engineers to hope there are no non-zero elements in the A(z) except the diagonal elements, since it will be easy to obtain rational transfer function matrix or Markov parameters of the system from such A(z) and B(z). We will do it now.

Assume the system is an initially relaxed one. Rewrite (2-8) as

$$\left[-B_w \; -B_{w-1} \; \cdots \; -B_0 \; A_w \; A_{w-1} \; \cdots \; A_0\right] \cdot \begin{bmatrix} U(k) \\ U(k+1) \\ \vdots \\ U(k+w) \\ Y(k) \\ Y(k+1) \\ \vdots \\ Y(k+w) \end{bmatrix} = 0_{p \times (N-w+1)} \cdot \qquad (3\text{-}8)$$

We use the same notation, but missing the superscript, therefore

$$C \cdot E_w = 0_{p \times (N-w+1)} \cdot \qquad (3\text{-}9)$$

In order to avoid those input-output difference descriptions in which there are some dependent relationships of outputs on the future inputs and save computation, we suggest to use a recursive procedure. At first define

$$F \triangleq \left[E_0 \,\vdots\, D \right] \qquad (3\text{-}10)$$

where

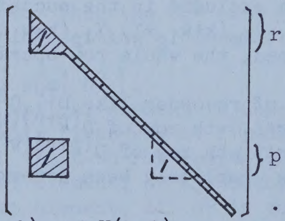
$$E_0 = \begin{bmatrix} U(k) \\ 0 \\ \vdots \\ Y(k) \\ 0 \\ \vdots \end{bmatrix} \qquad (3\text{-}11)$$

and D is a compatible identity matrix. Then the same row-operation will be taken, but there is a difference between these two algorithms. In this one there is no row-operation among those rows which correspond to the different components of output. After (p+r-1) steps of row-operation U(k+1) and Y(k+1) are added to E_0 as follows

$$E_1 = \left[\begin{array}{ccccccccccc|l}
X & X & X & X & \boxtimes & X & X & X & X & X & \cdots & \\
\boxtimes & X & X & X & 0 & X & X & X & X & X & \cdots & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & r \\
0 & X & X & X & 0 & X & X & X & X & X & \cdots & \\
\multicolumn{11}{c}{U(k+1)} & \\
\multicolumn{11}{c}{0} & \\
\multicolumn{11}{c}{\vdots} & \\
0 & X & \boxtimes & X & 0 & X & X & X & X & X & \cdots & \\
0 & X & X & X & 0 & X & X & \boxtimes & X & X & \cdots & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & & p \\
0 & X & X & X & 0 & X & X & X & X & X & \cdots & \\
\multicolumn{11}{c}{Y(k+1)} & \\
\multicolumn{11}{c}{0} &
\end{array}\right] \qquad \boxtimes \text{— maximal element} \qquad (3\text{-}12)$$

and

D =



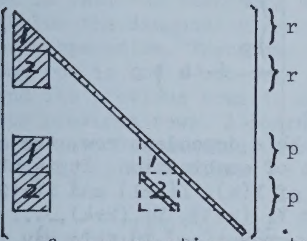
▨ — non-zero element area (3-13)

After that U(k+1) and Y(k+1) are compensated for previous (p+r-1) steps of row-operation, for example, E_1 and D are as follows respectively.

$$E_1 = \left[\begin{array}{l}
\left.\begin{array}{cccccccccccc} X&X&X&X&\boxtimes&X&X&X&X&X&\cdots \\ \boxtimes&X&X&X&0&X&X&X&X&X&\cdots \end{array}\right\} r \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\left.\begin{array}{cccccccccccc} 0&X&X&X&0&X&X&X&X&X&\cdots \\ 0&X&X&X&0&X&X&X&X&X&\cdots \end{array}\right\} r \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\qquad\qquad 0 \\
\qquad\qquad \vdots \\
\left.\begin{array}{cccccccccccc} 0&X&\boxtimes&X&0&X&X&X&X&X&\cdots \\ 0&X&X&X&0&X&X&\boxtimes&X&X&\cdots \end{array}\right\} p \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\left.\begin{array}{cccccccccccc} 0&X&0&X&0&X&X&X&X&X&\cdots \\ 0&X&X&X&0&X&X&0&X&X&\cdots \end{array}\right\} p \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\qquad\qquad 0 \\
\qquad\qquad \vdots
\end{array}\right] \qquad (3\text{-}14)$$

and

D =



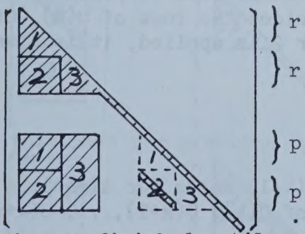
(3-15)

The next steps of row-operation are taken. E_1 and D are as follows and U(k+2) and Y(k+2) are added to E_1.

$$E_2 = \left[\begin{array}{l}
\left.\begin{array}{cccccccccccc} X&X&X&X&\boxtimes&X&X&X&X&X&\cdots \\ \boxtimes&X&X&X&0&X&X&X&X&X&\cdots \end{array}\right\} r \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\left.\begin{array}{cccccccccccc} 0&X&X&\boxtimes&0&X&X&X&X&X&\cdots \\ 0&X&X&0&0&X&X&X&\boxtimes&X&\cdots \end{array}\right\} r \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\qquad\qquad U(k+2) \\
\qquad\qquad 0 \\
\qquad\qquad \vdots \\
\left.\begin{array}{cccccccccccc} 0&X&\boxtimes&0&0&X&X&X&0&X&\cdots \\ 0&X&X&0&0&X&X&X&0&\boxtimes&\cdots \end{array}\right\} p \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\left.\begin{array}{cccccccccccc} 0&X&0&0&0&\boxtimes&X&X&0&X&\cdots \\ 0&X&X&0&0&X&X&\boxtimes&0&0&\cdots \end{array}\right\} p \\
\begin{array}{cccccccccccc} \vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots&\vdots& \end{array} \\
\qquad\qquad Y(k+2) \\
\qquad\qquad 0 \\
\qquad\qquad \vdots
\end{array}\right] \qquad (3\text{-}16)$$

and

D =

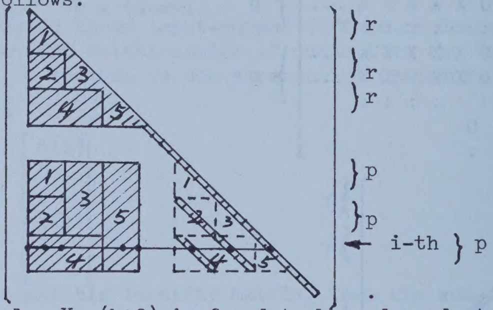


(3-17)

This process is not finished until p dependent rows are found.

It should be noted that after adding the rows of U(k+1) to the rows of

$Y(k)$ and $Y(k+1)$, and adding the rows of $U(k+2)$ to the rows of $Y(k)$, $Y(k+1)$ and $Y(k+2)$,...., some additional row-operation of adding the rows of $Y(k)$, $Y(k+1)$, $Y(k+2)$,.... to the rows below them are necessary and, in general speaking, the location variety of maximal element may happen, for example, there is the location variety of maximal element between (3-14) and (3-16). It means that there are more than one time of row-opeartion between the rows of $Y_i(k)$, $Y_i(k+1)$,..., $i = 1,\dots,p$. But the row-operation are linear, the final results are unique no matter how many times they are. D records the row-operation process, for example, after the row-operation on E_2 is finished D is as follows.

D =  (3-18)

If, for example, $Y_{i_1}(k+2)$ is found to be a dependent row, the correspondent row of D just records the coefficients of combination. It is obvious that $Y_{i_1}(k+2)$ is not dependent on the rows of $Y(k)$, $Y(k+1)$ and $Y(k+2)$ except $Y_{i_1}(k)$ and $Y_{i_1}(k+1)$. On the other hand $Y_{i_1}(k+3)$, $Y_{i_1}(k+4)$,.... will be excluded in the succedent steps of row-operation. In this way it can be expected that there are no non-zero elements except diagonal in the $A(z)$. As assumed above the system is initially relaxed one, therefore it is easy to find rational transfer function matrix or Markov parameters from such pair of $A(z)$ and $B(z)$.

Of course, Algorithm 1 also has recursive form itself, which is easier than this one.

4. PERSISTENTLY EXCITING INPUT

As mentioned in the Algorithm 1 of the last section, input sequence should be persistently exciting. But what is the measurement of that the input sequence is persistently exciting enough to our problem.

It is obvious from (2-8) that if all rows of $U(k)$, $U(k+1)$,...,$U(k+w)$ are linear independent on each other and $Y_1(k+w_1)$, $Y_2(k+w_2)$,...,$Y_p(k+w_p)$ are linear dependent on their previous rows respectively, where w_1, w_2,...,w_p are the smallest ones of the integers which have this property, A_0, B_0, A_1, B_1,...,A_w, B_w can be uniquely determined by row-operation introduced in this paper.

About Algorithm 2 we can say the same words except the only difference that each of $Y_1(k+w_1)$, $Y_2(k+w_2)$,..., $Y_p(k+w_p)$ is dependent on not all its previous rows, but only those which are the same components of $Y_1(k+w_1)$, $Y_2(k+w_2)$,...,$Y_p(k+w_p)$ respectively besides the rows of $U(k)$, $U(k+1)$,..., $U(k+w)$. Therefore while Algorithm 1 or 2 is applied, it is always assumed that

$$\mathrm{RANK}\begin{bmatrix} U(k) \\ U(k+1) \\ \vdots \\ U(k+w) \end{bmatrix} = (p+r)(w+1) \qquad (4\text{-}1)$$

It is easy to check whether the (4-1) holds for Algorithm 2. While the (4-1) does not hold, at least, a row of $U(k)$, $U(k+1)$,..., $U(k+w)$ will be turned to zeros in the row-operation.

The case is a little different in the Algorithm 1. Of course, while the

(4-1) does not hold, the same thing will certainly happen like Algorithm 2. But it is possible that even though the (4-1) holds, a row of $U(k)$, $U(k+1)$, ..., $U(k+w)$ is turned to zeros before the p dependent rows $Y_1(k+w_1), \ldots, Y_p(k+w_p)$ are found. It leads either some dependent relationships of outputs on the future inputs or some dependence on the non-future inputs, but the coefficients to be not unique. Both of them are useless. Fortunately, the probability of that the case mentioned above will happen is nearly zero, especially, as the length of input/output measurements N is large enough.

5. BASIC ALGORITHMS' EXTENSION

At first we discuss the extension of Algorithm 1. According to (2-14), it is true that

$$C \cdot (E_w \cdot E_w') = 0_{p \times p} . \tag{5-1}$$

We can take the same row-operation on (5-1) to determine the matrix C, while input/output measurement sequences (2-1) and (2-2) are given. But there are two points different from what is introduced in section 3.

The first is that the searching maximal element is not necessary. Just the elements below the diagonal element of each row are turned into zeros in each step of row-operation. Therefore the second is that if the diagonal element of some row is not a non-zero element, i.e. the main minor included of this row and its previous rows is equal to zero, then this row is dependent on all its previous rows. A correspondent row of C has been determined.

In order to save computation, a recursive procedure is introduced here. In (5-1) $E_w \cdot E_w'$ is as follows

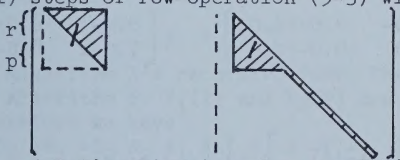
$$E_w \cdot E_w' = \left[\begin{array}{cc:cc:c} \multicolumn{2}{c:}{E_0 \cdot E_0'} & \multicolumn{2}{c:}{E_1 \cdot E_1'} & \\ U(k)U'(k) & U(k)Y'(k) & U(k)U'(k+1) & \cdots\cdots & \cdots \\ Y(k)U'(k) & Y(k)Y'(k) & Y(k)U'(k+1) & \cdots\cdots & \cdots \\ \hdashline U(k+1)U'(k) & U(k+1)Y'(k) & U(k+1)U'(k+1) & \cdots\cdots & \cdots \\ Y(k+1)U'(k) & Y(k+1)Y'(k) & Y(k+1)U'(k+1) & Y(k+1)Y'(k+1) & \cdots \\ \hdashline \vdots & \vdots & \vdots & \vdots & \\ \vdots & \vdots & \vdots & \vdots & \end{array}\right] \tag{5-2}$$

At first, define

$$F = \left[E_0 \cdot E_0' \;\vdots\; D \right] = \left[\begin{array}{cc:c} U(k)U'(k) & U(k)Y'(k) & \\ Y(k)U'(k) & Y(k)Y'(k) & D \end{array} \right] . \tag{5-3}$$

After (p+r-1) steps of row-operation (5-3) will be turned into

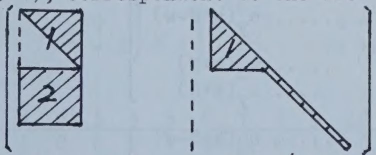
F =



(5-4)

Put the part of E_1E_1', which is below the E_0E_0' in the (5-2) on the suitable location of (5-4), correspondent to the area 2 of (5-5) as follows

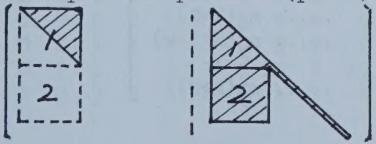
F =



(5-5)

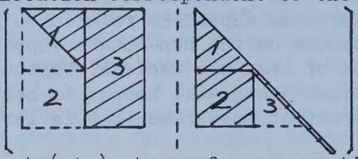
Compensate to this part for previous (p+r-1) steps of row-operation.

F =



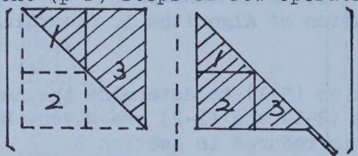
(5-6)

Then pre-multiply the submatrix consisted of the right-half of $E_1 \cdot E_1'$ in the (5-2) by the submatrix consisted of the areas 1,2 and 3 of D. Put the product on suitable location correspondent to the area 3 of (5-7) as follows

F = 

(5-7)

Proceed to next (p+r) steps of row-operation. The result is as follows

F = 

(5-8)

This process is not finished until the p dependent rows are found.

Algorithm 2 also can be extended in the same way. We have to cut out the extension of Algorithm 2, because the length of this paper is limited.

6. THE CASE IN THE PRESENCE OF NOISE

In this section, it will be considered that there are some uncorrelated noise with zero mean in the measuring input and output.

The measurement sequences of noisy input and output will be

$$u^*(k),\ u^*(k+1),\ u^*(k+2), \ldots\ldots \tag{6-1}$$

$$y^*(k),\ y^*(k+1),\ y^*(k+2), \ldots\ldots \tag{6-2}$$

where

$$u^*(k+m) = u(k+m) + n_u(k+m) \qquad m = 0,1,\ldots.,N$$

$$y^*(k+m) = y(k+m) + n_y(k+m) \qquad m = 0,1,\ldots.,N$$

E_w^* is defined as

$$E_w^* = \begin{bmatrix} U^*(k) \\ Y^*(k) \\ : \\ U^*(k+w) \\ Y^*(k+w) \end{bmatrix}, \tag{6-3}$$

if Algorithm 1 is used, or as

$$E_w^* = \begin{bmatrix} U^*(k) \\ : \\ U^*(k+w) \\ Y^*(k) \\ : \\ Y^*(k+w) \end{bmatrix}, \tag{6-4}$$

if Algorithm 2 is used. So,

$$E_w^* = E_w + N_w \tag{6-5}$$

where E_w is defined as (2-13) or (3-8) and N_w is defined as

$$N_w = \begin{bmatrix} n_u(k) \ldots\ldots & \ldots\ldots n_u(k+N-w) \\ n_y(k) \ \ldots\ldots & \ldots\ldots n_y(k+N-w) \\ : & : \\ n_u(k+w) \ldots\ldots & \ldots\ldots n_u(k+N) \\ n_y(k+w) \ldots\ldots & \ldots\ldots n_y(k+N) \end{bmatrix} \tag{6-6}$$

or

$$N_w = \begin{bmatrix} n_u(k) \ \ \ldots\ldots & \ldots\ldots n_u(k+N-w) \\ : & : \\ n_u(k+w) \ldots\ldots & \ldots\ldots n_u(k+N) \\ n_y(k) \ \ \ldots\ldots & \ldots\ldots n_y(k+N-w) \\ : & : \\ n_y(k+w) \ldots\ldots & \ldots\ldots n_y(k+N) \end{bmatrix} \tag{6-7}$$

Therefore

$$\begin{aligned} E_w^* \cdot E_w^{*\prime} &= (E_w+N_w)(E_w+N_w)' \\ &= E_w \cdot E_w' + E_w \cdot N_w' + N_w \cdot E_w' + N_w \cdot N_w' \\ &= E_w \cdot E_w' + (N-w+1)\left[\frac{E_w \cdot N_w'}{N-w+1} + \frac{N_w \cdot E_w'}{N-w+1} + \frac{N_w \cdot N_w'}{N-w+1}\right] \end{aligned}$$

We assume N is large enough, so

$$\frac{E_w \cdot N_w'}{N-w+1} \approx 0 \qquad (6\text{-}8)$$

$$\frac{N_w \cdot E_w'}{N-w+1} \approx 0 \qquad (6\text{-}9)$$

Since the noise is not correlated with the input/output sequence and

$$\frac{N_w \cdot N_w'}{N-w+1} = V_N \qquad (6\text{-}10)$$

where V_N is the covariance matrix of any column of (6-6) or (6-7). Then we have

$$E_w^* \cdot E_w^{*\prime} = E_w \cdot E_w' + (N-w+1)V_N$$

or

$$E_w \cdot E_w' = E_w^* \cdot E_w^{*\prime} - (N-w+1)V_N \quad . \qquad (6\text{-}11)$$

$E_w^* \cdot E_w^{*\prime}$ can be evaluated and V_N is assumed to be known. Substituting (6-11) into (5-1) and using extension's Algorithm 1 or 2, the solution of the identification problem in the presence of noise can be obtained.

ACKNOWLEDGMENT

The author wishes to express his heartfelt gratitude to Professor C. T. Chen of SUNY at Stony Brook for his giving very nice advice and encouragement.

APPENDIX : NUMERICAL EXAMPLES

Example 1 .

m	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$u_1(m)$	1	0	0	1	0	0	1	0	0	1	0	0	1	0
$u_2(m)$	1	0	0	0	1	0	0	0	1	0	0	0	1	0
$y_1(m)$	0	2	0	2	1	2	2	3	1	4	3	3	3	5
$y_2(m)$	1	2	3	3	3	5	6	5	6	8	8	8	9	10

a. Use Algorithm 1. Assume $w^* = 2$. $Y_1(1)$ and $Y_2(1)$ are found to be dependent, i.e. $w_1=w_2=1$. The elements on the 7 and 8-th row of D are the coefficients. Therefore we have

$$\begin{bmatrix} -B_1 & A_1 & -B_0 & A_0 \end{bmatrix} = \left[\begin{array}{cc|cc|cc|cc} -0.6 & -0.8 & 0.6 & -0.6 & -0.2 & 0.2 & 1.0 & 0 \\ 0.8 & -2.1 & -0.8 & -0.7 & -0.9 & -0.1 & 0 & 1.0 \end{array}\right] .$$

It means that

$$\begin{bmatrix} z+0.6 & -0.6 \\ -0.8 & z-0.7 \end{bmatrix} y = \begin{bmatrix} 0.2z+0.6 & -0.2z+0.8 \\ 0.9z-0.8 & 0.1z+2.1 \end{bmatrix} u$$

b. Use Algorithm 1's recursive form. The same result is got.

c. Use Algorithm 2. $Y_1(2)$ and $Y_2(2)$ are found to be dependent, i.e. $w_1=w_2=2$. Therefore we have

$$\begin{bmatrix} -B_2 & -B_1 & -B_0 & A_2 & A_1 & A_0 \end{bmatrix} = \left[\begin{array}{cc|cc|cc|cc|cc|cc} 1 & -1 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & -2 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{array}\right] .$$

It means that

$$\begin{bmatrix} z^2-1 & 0 \\ 0 & z^2-1 \end{bmatrix} y = \begin{bmatrix} z-1 & z+1 \\ z^2 & 2z+2 \end{bmatrix} u \quad .$$

Example 2 (k = 1) .

m	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$u_1(m)$	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
$u_2(m)$	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
$u_3(m)$	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
$y_1(m)$	0	1	1	1	1	1	2	2	2	2	2	3	3	3	3
$y_2(m)$	0	2	1	4	3	7	5	10	8	15	11	20	16	25	20
$y_3(m)$	1	2	3	2	4	2	6	4	7	4	7	6	10	6	10

m	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
$u_1(m)$	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
$u_2(m)$	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
$u_3(m)$	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
$y_1(m)$	3	4	4	4	4	4	5	5	5	5	5	6	6	6	6
$y_2(m)$	31	27	39	32	45	39	55	47	63	55	72	64	83	74	94
$y_3(m)$	6	13	8	13	8	14	10	16	10	17	10	19	12	20	12

a. Use Algorithm 1's extension. $Y_1(2)$, $Y_2(3)$ and $Y_3(3)$ are found to be dependent, i.e. $w_1=1$, $w_2=w_3=2$. The coefficients are as follows

$$[-B_2\ A_2\ -B_1\ A_1\ -B_0\ A_0]$$

$$= \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc} 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 9 & -1 & 0 & 1 & 4 & 3 & -1 & 0 & 0 & -2 & -4 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & 0 & 0 & -1 & 0 & -2 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \end{array}\right].$$

It means that

$$\begin{bmatrix} z-1 & 0 & 0 \\ 0 & z^2-2z+1 & -4z+4 \\ 0 & 0 & z^2-1 \end{bmatrix} y = \begin{bmatrix} 0 & 1 & 0 \\ -3z+1 & z-9 & 1 \\ z^2 & 2z+2 & 0 \end{bmatrix} u .$$

b. Use Algorithm 2's extension. In E_1E_1', $Y_1(2)$ is found to be dependent, i.e. $w_1=1$. In E_2E_2', $Y_3(3)$ is found to be dependent, i.e. $w_3=2$. In E_3E_3', $Y_2(4)$ is found to be dependent, i.e. $w_2=3$. The coefficients are as follows

$$[-B_3\ -B_2\ -B_1\ -B_0\ A_3\ A_2\ A_1\ A_0]$$

$$= \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc|ccc|ccc} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 2 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & -2 & 0 & 0 & -2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{array}\right].$$

It means that

$$\begin{bmatrix} z-1 & 0 & 0 \\ 0 & z^3-z^2-z+1 & 0 \\ 0 & 0 & z^2-1 \end{bmatrix} y = \begin{bmatrix} 0 & 1 & 0 \\ z^2-2z+1 & z^2-1 & z+1 \\ z^2 & 2z+2 & 0 \end{bmatrix} u$$

or

$$y/u = \begin{bmatrix} 0 & 1/z-1 & 0 \\ 1/z+1 & 1/z-1 & 1/(z-1)^2 \\ z^2/z^2-1 & 2/z-1 & 0 \end{bmatrix} .$$

REFERENCES

1. C. T. Chen, "Identification of linear-invariant multivariable system", (to be published).
2. R. Guidorze, "Canonical structures in the identification of multivariable systems", Automatica, vol. 11, pp. 361-374, 1975.
3. W. A. Wolovich, "The determination of state-space representations for linear multivariable systems", Automatica, vol. 9, pp. 97-106, 1973.
4. D. T. Mital and C. T. Chen, "Irreducible canonical form realization of a rational matrix", Int. J. Contr. vol. 18, pp. 881-887,1973.
5. S. P. Bingulac and M. A. C. Farias, "Identification and minimal realization of multivariable systems", Proc. 4th IFAC Symposium (Editor:D. P. Atherton), pp. 373-378, 1977.
6. S. Beghelli and R. Guidorse, "A new input-output canonical form for multivariable systems", IEEE Trans. Automat. Contr. vol.21,pp 692-696, 1976.
7. C. T. Chen, Introduction to Linear System Theory, New York: Holt, Rinehart and Winston, 1970.
8. W. A. Wolovich, Linear Multivariable Systems, New York: Springer-verlag ,1974.
9. P. Eykhoff, System Identification, New York: Wiley, 1974.
10. T. C. Hsia, System Identification, Lexington, Mass.: Lexington, 1977.
11. T. Kailath, Linear Systems, Englewood Cliffs, N. J.: Prentice-Hall, 1980.

ROUTH-HURWITZ APPROXIMATION OF MULTIVARIABLE SYSTEMS

N.K. SINHA, I. EL-NAHAS and R.T.H. ALDEN
Group on Simulation, Optimization and Control
Faculty of Engineering, McMaster University
Hamilton, Ontario

ABSTRACT

A simple procedure is presented for obtaining reduced-order models for linear multivariable systems. This method does not require the calculation of the eigenvalues and eigenvectors of the system matrix, and hence it is computationally very attractive. Furthermore, the stability of the low-order model is guaranteed if the original system is stable. The proposed method is a generalization of the Routh approximation method of Hutton and Friedland which cannot be easily applied to multivariable systems. The details of the proposed approach are illustrated by means of examples.

INTRODUCTION

Although a number of different methods are available for obtaining reduced-order models of single-input systems, the Routh approximation method originally proposed by Hutton and Friedland [1] has two major advantages. It does not require the determination of the eigenvalues of the system, and the stability of the reduced model is guaranteed if the original system is stable. Hence, it is considered more powerful than methods retaining the dominant eigenvalues [2-4], since these require the computation of the eigenvalues, and the continued fraction expansion method of Chen and Shieh [5] which, although computationally simple, does not guarantee a stable reduced-order model even if the original system is stable.

Attempts to extend the Routh approximation method to the more general case of multivariable systems [6] have not proved so successful from the computational point of view. Other methods for reducing multivariable systems are even less attractive computationally since they require the computation of the eigenvalues and eigenvectors of a large system matrix [7].

The paper is an attempt to extend the Routh approximation method to the case of multivariable systems. First the method of Hutton and Friedland will be presented in a new light which is not only more straightforward than their original derivation[1], but is also readily extended to the multivariable case. Although this approach requires that the state equations of the system be transformed to either the column-companion or the row-companion form [8], this transformation is quite straightforward and can be performed in a routine manner without any matrix inversion [9]. The proposed procedure for the reduction of multivariable systems will be discussed in detail, and will be illustrated with two examples.

STATEMENT OF THE PROBLEM

Consider a linear time-invariant system described by the equations

$$\dot{x} = Ax + Bu$$

$$y = Cx \tag{1}$$

where $x \in R^n$, $y \in R^m$ and $u \in R^p$ are the state, output and input vectors respectively.

The objective of model reduction is to obtain the system equations

$$\dot{x}_r = F\, x_r + G\, u$$

$$y_r = H\, x_r \tag{2}$$

where $x_r \in R^r$ and $r << n$ so that $y_r \in R^m$ is, in some defined sense, a close approximation to y for all inputs $u \in \Omega$, a defined set of admissible inputs.

Alternatively, the system, for zero initial state, may be described by its transfer function matrix G(s), defined through the equation

$$Y(s) = G(s)\ U(s) \tag{3}$$

In this case, the reduced-order transfer function matrix $G_r(s)$ should be such that

$$Y_r(s) = G_r(s)\ U(s) \tag{4}$$

and should be a close approximation to Y(s) in the time domain.

We shall first consider a solution to this problem for the single-input single-output case, which can be easily applied to either of the two representations. This will then be generalized to the multivariable case.

ROUTH APPROXIMATION METHOD FOR THE SINGLE-INPUT SINGLE-OUTPUT CASE

In their original work on the Routh approximation method [1], Hutton and Friedland have described their method through transformation to the α-β canonical form, based on the truncation of the Routh table for the denominator polynomial of the transfer function. An alternative way of looking at their method is more revealing and can be extended to the multivariable case in a straightforward manner.

We start with the continued fraction expansion about the origin of the ratio of the even and odd parts of the denominator polynomial. Truncation of this expansion gives the denominator of the lower order transfer function, and guarantees stability through the Hurwitz criterion provided that the original system is stable. The coefficients of the numerator polynomial are then obtained by equating r time moments of the two models, where r is the order of the reduced system to provide a partial Padé approximation to the Taylor's series expansion of G(s). The following simple example, taken from the work of Hutton and Friedland [1] will illustrate the proposed method. Consider

$$G(s) = \frac{12s^3 + 528s^2 + 1440s + 4320}{2s^4 + 36s^3 + 204s^2 + 360s + 240} \tag{5}$$

From the denominator polynomial we get

$$H_4(s) = \frac{240 + 204s^2 + 2s^4}{360s + 36s^3} = \frac{2}{3s} + \cfrac{1}{\cfrac{2}{s} + \cfrac{1}{\cfrac{45}{8s} + \cfrac{1}{\cfrac{16}{s}}}} \tag{6}$$

as the continued fraction expansion about s = 0.

If we desire a second-order model, we retain the first two terms only. Hence,

$$H_2(s) = \frac{2}{3s} + \frac{1}{\frac{2}{s}} = \frac{4+3s^2}{6s} \tag{7}$$

Therefore, the reduced-order transfer function will be of the form

$$G_r(s) = \frac{a_1 s + a_o}{3s^2 + 6s + 4} \tag{8}$$

and we have to determine a_1 and a_0 so that the first two time moments of $G(s)$ and $G_r(s)$ are identical.

The time moments of a transfer function $G(s)$ are defined as

$$T_i \triangleq \int_0^{\infty} t^i \, g(t) \, dt \tag{9}$$

where $g(t)$ is the inverse Laplace transform of $G(s)$. However, it is not necessary to obtain the time moments by the integration indicated in equation (9). If we expand $G(s)$ in the Taylor's series to obtain

$$G(s) = g_0 + g_1 s + g_2 s^2 + \ldots \tag{10}$$

then it can easily be shown that

$$T_i = (-1)^i \, i! \, g_i \tag{11}$$

Hence, to match time moments, we first rewrite the original transfer function in the ascending powers of s as

$$G(s) = \frac{4320 + 1440s + 528s^2 + 12s^3}{240 + 360s + 204s^2 + 36s^3 + 2s^4} \tag{12}$$

Performing long division, we obtain the Taylor series expansion

$$G(s) = 18 - 21s + 18.4s^2 - 12.4s^3 + \ldots\ldots. \tag{13}$$

Since we have only two degrees of freedom (choice of a_0 and a_1), we can match only two terms. The first two time moments are matched by making $a_1 = 24$ and $a_0 = 72$, as determined by multiplying the denominator of equation (4) by 18-21s, and then equating the corresponding coefficients

Hence, $$G_r(s) = \frac{24s + 72}{3s^2 + 6s + 4} , \tag{14}$$

which is exactly the same reduced-order transfer function as obtained by Hutton and Friedland.

It may be pointed out that although the method of Hutton and Friedland gives a partial Pade approximation to the Taylor's series expansion of G(s), the proposed approach provides more flexibility in that instead of matching time moments, we could match the Markov parameters, or a combination of the two. It may be recalled that the Markov parameters are defined as

$$J_i = C A^i B \tag{15}$$

and are obtained from the expansion of G(s) in negative powers of s, i.e.,

$$G(s) = C(sI-A)^{-1} B = \sum_{i=0}^{\infty} J_i s^{-i-1} \tag{16}$$

In general, we may say that matching time-moments makes the steady-state response of the reduced model closer to that of the original, whereas matching the Markov parameters improves the transient response. For instance, in the example considered here, we could have matched J_0 and T_0 instead of T_0 and T_1. The resulting transfer function is of the same form as (8), but now the coefficients of the numerator are adjusted to match J_0 and T_0. Since J_0 is easily found as 6, the resulting reduced-order transfer function is obtained as

$$G_r'(s) = \frac{18s + 72}{3s^2 + 6s + 4} . \tag{17}$$

The responses of G(s), $G_r(s)$ and $G_r'(s)$ to a unit step are shown in Fig. 1. It will be seen that the reduced-order models give responses close to that of the actual system, and that the response of $G_r(s)$ is a closer approximation in this case.

ROUTH APPROXIMATION OF MULTIVARIABLE SYSTEMS

The procedure described in the previous section can be readily extended to multivariable systems. As such systems are generally described by state equations, the first step is to transform these equations to the column- or row-companion form [8]. This transformation can be carried out in a routine manner without matrix inversion [9], and the authors have a working computer program for this purpose [10]. If the system is described in the form of the transfer function matrix, it is possible to obtain a minimal state-space realization directly in the canonical form [11].

In either the column-companion or the row-companion form, the A-matrix for the system is block triangular. Hence, the characteristic polynomial of each block is obtained separately and directly from the last column (or row) of the block. Also, either the B-matrix (for the column-companion form) or the C-matrix (for the row-companion form) consists of unit vectors which are known directly from the structure of A. For the reduction process, we determine the reduced characteristic polynomial for each block using the continued-fraction expansion as in the previous section. The remaining unknown elements of the matrices are then obtained so as to match the desired time moments and/or Markov parameters. We shall illustrate the procedure by means of an example.

Consider a two-input two-output system which has been brought to the column-companion form, and is described by the following matrices.

$$A = \left[\begin{array}{ccccccc} 0 & 0 & 0 & -80 & 0 & 0 & -20 \\ 1 & 0 & 0 & -100 & 0 & 0 & -20 \\ 0 & 1 & 0 & -62 & 0 & 0 & -10 \\ 0 & 0 & 1 & -12 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -80 \\ 0 & 0 & 0 & 0 & 1 & 0 & -96 \\ 0 & 0 & 0 & 0 & 0 & 1 & -17 \end{array}\right] \qquad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 2 & -1 & 0 & 1 & 1 \\ 0 & 2 & 0 & 1 & 2 & 0 & -1 \end{bmatrix} \tag{18}$$

The characteristic polynomials of the two blocks are obtained immediately as

$$P_1(s) = s^4 + 12s^3 + 62s^2 + 100s + 80 \tag{19}$$

and

$$P_2(s) = s^3 + 17s^2 + 96s + 80 \tag{20}$$

Reducing this to the second- and first-order, respectively, through the continued fraction method described in the previous section, we get

$$\hat{P}_1(s) = s^2 + \frac{250}{131}s + \frac{200}{131} \tag{21}$$

and

$$\hat{P}_2(s) = s + \frac{5}{6} \tag{22}$$

Hence, we get

$$F = \begin{bmatrix} 0 & -\frac{200}{131} & f_1 \\ 1 & -\frac{250}{131} & f_2 \\ 0 & 0 & -5/6 \end{bmatrix} \qquad G = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{23}$$

and $$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix}$$

Since we have eight unknown parameters in F and H, we can match the time moments T_1 and T_2 of the system and the reduced-order model. Due to the canonical forms of A and F, they are easily inverted, and we have

$$A^{-1} = \begin{bmatrix} -1.25 & 1 & 0 & 0 & 0.0625 & 0 & 0 \\ -0.775 & 0 & 1 & 0 & 0.06875 & 0 & 0 \\ -0.15 & 0 & 0 & 1 & 0.0375 & 0 & 0 \\ -0.0125 & 0 & 0 & 0 & 0.003125 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.2 & 1 & 0 \\ 0 & 0 & 0 & 0 & -0.2125 & 0 & 1 \\ 0 & 0 & 0 & 0 & -0.0125 & 0 & 0 \end{bmatrix} \tag{24}$$

$$F^{-1} = \begin{bmatrix} -1.25 & 1 & 1.2f_2-1.5f_1 \\ -0.655 & 0 & -0.786f_1 \\ 0 & 0 & -1.2 \end{bmatrix} \tag{25}$$

Equating the time moments $T_1 = CA^{-1}B = HF^{-1}G$, and $T_2 = CA^{-2}B = HF^{-2}G$, we get eight equations for eight unknowns. The solution of these equations is easy as they are partly decoupled and the largest matrix inverted is of dimension four. The resulting matrices F and H are given below

$$F = \begin{bmatrix} 0 & -\frac{200}{131} & -0.675356 \\ 1 & -\frac{250}{131} & -0.47595 \\ 0 & 0 & -5/6 \end{bmatrix} \quad H = \begin{bmatrix} 1.2166 & 0.025566 & 0.5348337 \\ 0.457404 & 1.512586 & 2.709936 \end{bmatrix} \tag{26}$$

If instead of matching T_1 and T_2, we match $J_0 = CB = HG$ and T_1, we get

$$F = \begin{bmatrix} 0 & -\frac{200}{131} & -0.0818 \\ 1 & -\frac{250}{131} & -0.20146 \\ 0 & 0 & -5/6 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0.438931297 & 0 \\ 0 & 2.85496183 & 2 \end{bmatrix} \qquad (27)$$

For the sake of comparison, another reduced-order model for the system was obtained using aggregation with partial realization [7], matching the time moments T_1 and T_2. The resulting matrices are given below

$$F' = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 1 & -1 \end{bmatrix} \qquad G' = \begin{bmatrix} 0 & 0.577 \\ 0.25 & 0 \\ 0.25 & 0 \end{bmatrix}$$

$$H' = \begin{bmatrix} 0.15706239 & -9.157 & 6.15 \\ 3.894064 & -17.1239 & 6.25 \end{bmatrix} \qquad (28)$$

It is interesting to note that the eigenvalues of the Routh approximation model are $-(\frac{5}{6})$ and $0.954 \pm j0.785$, whereas the eigenvalues in the aggregated model are -1 and $-1 \pm j1$.

The responses of the first output of the seventh-order system and the two reduced models to a unit step applied to the first input are shown in Fig. 2. It is seen that while the three responses have identical steady-state values (since T_1 has been matched), there is a small deviation in the initial part. This is to be expected since the fast transients have been neglected in the reduced models. The response of the Routh approximation model is a little closer to that of the original system. Perhaps the reason is that by removing the constraint that the eigenvalues of the reduced model be identical to those of the original system, we get an improved approximation. Also, the amount of computation required for obtaining the aggregated model matching the time moments is considerably greater.

CONCLUSIONS

The Routh approximation method for multivariable systems presented in this paper is computationally much superior to the methods based on aggregation since we do not have to calculate the eigenvalues and eigenvectors of the large system matrix [7]. The method of partial realizations [13] requires less computation, but sometimes the reduced-order model turns out to be unstable even though the original system may be stable [12]. By combining the truncation of the continued-fraction expansion of the characteristic polynomial of the system with matching time moments and/or Markov parameters, it is possible to obtain stable low-order models with responses close to that of the original system.

It may be pointed out that the proposed approach is more flexible than that of Hutton and Friedland [1] even for the single-input single- output case. In the latter method, the degree of the numerator polynomial is constrained to be one less than that of the denominator. This may not always be desirable, especially when the low-order model is used for compensator design using frequency-response techniques since the phase-shift of the model at high frequencies may be quite different from that of the original system if the difference between the degrees of the numerator and the denominator is not maintained. The proposed method is more flexible in that one is free not only to select the degree of the numerator, but also to match a combination of time moments and Markov parameters. This flexibility is retained with multivariable systems.

One weakness of the method is that the multivariable system must first be transformed to the column- or row-companion form. While this transformation is fairly straightforward and can be carried out in a routine manner, there may be some cases when the number of blocks is less than the number of inputs (or the number of outputs for the row-companion form). This situation varies if the system is completely controllable from only some of the inputs (or completely observable from only some of the outputs). In this case, the calculation of the system parameters matching the time-moments (or Markov parameters) becomes more involved. The same difficulty could also occur in obtaining aggregated models matching time moments [12].

A number of questions remain unanswered. For example, one would like to know what order of the reduced model will be most suitable. Hutton and Friedland [1] have proposed an approach based on the impulse energy, which can be calculated conveniently from their α-β canonical form for models of different order. It would be desirable to develop a similar criterion for the multivariable case. This is currently being investigated. Furthermore, the usefulness of a low-order model should not be judged only by comparing responses to certain inputs. It may often be more important to see how well a particular low-order model may be utilized for the design of a controller for the original system. This is a topic for future investigation.

ACKNOWLEDGEMENTS

The support of this research by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

REFERENCES

1. Hutton, M.F. and Friedland, B., "Routh approximation for reducing order of linear time-invariant systems", IEEE Trans. on Automatic Control, vol. AC-20, 1975, pp. 329-337.

2. Davison, E.J., "A method for simplifying linear dynamic systems", ibid, vol. AC-11, 1966, pp. 93-101.

3. Aoki, M., "Control of large scale dynamic systems by aggregation", ibid, 1968, pp. 246-253.

4. Mitra, D., "On the reduction of complexity of linear dynamic models", Report AEEW-R520, United Kingdom Atomic Energy Authority, Winfrith, 1967.

5. Chen, C.F. and Shieh, L.S., "A novel approach to linear model simplification", International Journal of Control, vol. 8, 1968, pp. 561-570.

6. Hutton, M.F., "Routh approximations in state space", Proc. 8th. Annual Conf. on Modeling and Simulation (Pittsburgh, PA) 1977, pp. 311-315.

7. Hickin, J.D. and Sinha, N.K., "Model reduction for linear multivariable systems", IEEE Transactions on Automatic Control, vol. AC-25, (to be published in December 1980).

8. Sinha, N.K. and Rózsa, P., "Some canonical forms for linear multivariable systems", International Journal of Control, vol. 20, 1976, pp. 865-883.

9. Hickin, J.D. and Sinha, N.K., "On the transformation of linear multivariable systems to canonical forms", International Journal of Systems Science, vol. 10, 1979, pp. 783-796.

10. Li, N.H. and Sinha, N.K., "Computer program for transformation of linear multivariable systems to canonical forms", Report SOC-246, Faculty of Engineering, McMaster University, 1980.

11. Rózsa, P. and Sinha, N.K., "Minimal realization of a transfer function matrix in canonical forms", International Journal of Control, vol. 21, 1975, pp. 273-284.

12. Elrazaz, Z. and Sinha, N.K., "Modeling and simulation of power systems for dynamic analysis", Proc. IFAC Symp. on Computer Applications to Large Scale Power Systems, (New Delhi, India), 1979, pp. 204-211.

13. Hickin, J.D. and Sinha, N.K., "A new method for reducing multivariable systems", Proc. 7th. Annual Conference on Modeling and Simulation (Pittsburgh, PA) 1976, pp. 259-263.

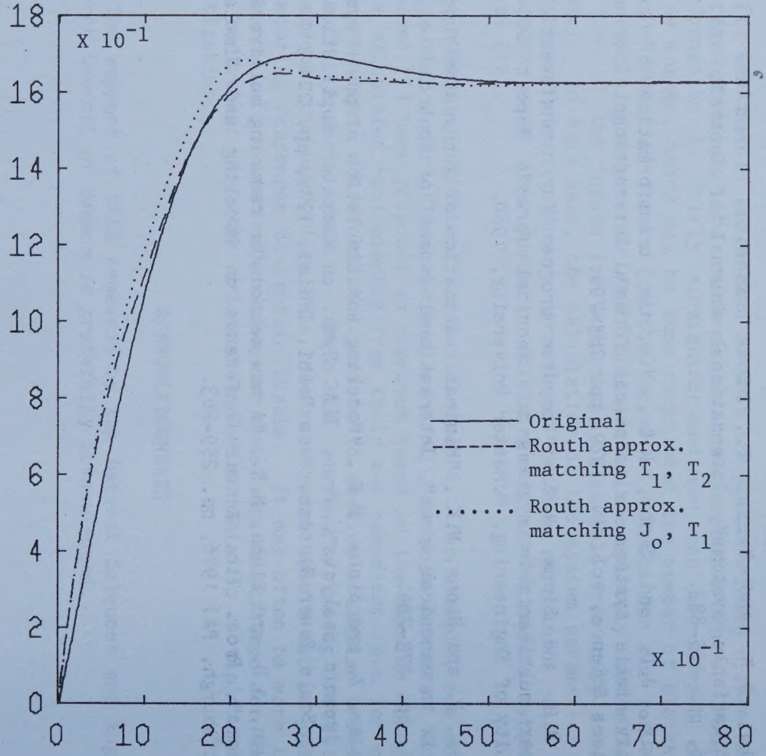


Fig. 1. Comparison of responses of single-input single-output system to a unit step.

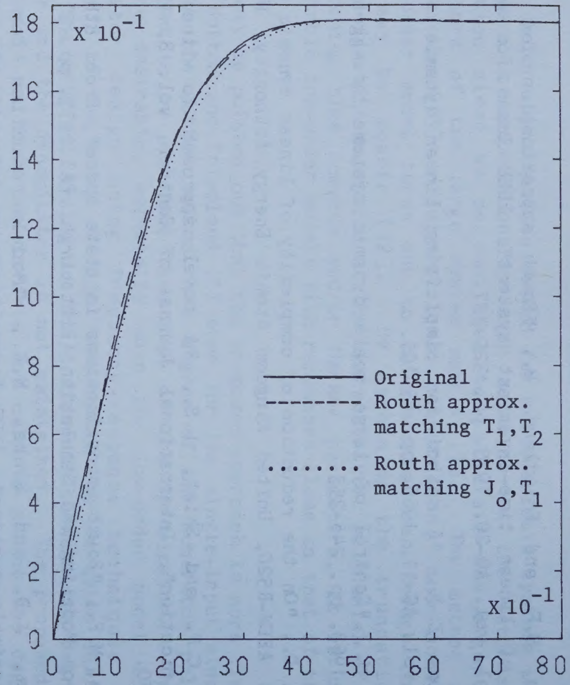


Fig. 2. Comparison of responses of the first output to unit step at the first input.

TRANSFER FUNCTION IDENTIFICATION FOR MULTI-VARIABLE LINEAR SYSTEMS

GUO-HONG XIA
SUNY, Stony Brook, New York and Chinese Academy of Space Technology, Peking

ABSTRACT

A direct procedure is presented for identifying the transfer function matrix of linear, time-invariant, discrete-time multivariable systems from noise-free input-output measurements (I/O). The identification of the parametric matrix can be obtained at one step by making a so-called characteristic subsystem. The determination of the degree of the characteristic polynomial is accomplished by using a recursive algorithm. Examples are included to show the operation of the procedure.

INTRODUCTION

System identification has attracted great attention during the past decade. Most of the work has been concerned with the identification of a system with single input and single output, while relatively little work has been done on the multi-variable systems. Ho and Kalman[1], Gopinath[2], Budin[3] and Silverman[4] has investigated the problem of identification of a state dynamic equation. Rosenbrok[5] has considered the realization of transfer function. Rissanen[6], Mital and Chen[7], Wu and Ruruta[8], Shinha and Kwong[9] has considered the identification of a transfer function matrix of multivariable systems. Most of the work, however, were concerned with the subsystem was single-output and multi-input; the identification of the whole system is accomplished in p separate steps (p is the number of subsystems). For such a procedure, the computational requirements are very large if p is large.

This paper develops an algorithm for identifying the transfer function of the multivariable systems at one step, making a characteristic subsystem to identify the least common denominator of the transfer function of the overall systems. The proposed algorithm is computationally possibly simpler than the existing identification techniques.

PROBLEM STATEMENT

Let's consider the difference equation representation of multivariable time-invariant linear discrete systems. The general nth order difference equation relating inputs u(k) and ouputs y(k) is

$$y(k+n) + a_1 y(k+n-1) + a_2 y(k+n-2) + \dots + a_n y(k) = m_1 u(k+n-1) + m_2 u(k+n-2) + \dots + m_n u(k) \quad (1)$$

where k is the integer time index, y(k) and u(k) are p × 1 and r × 1 vectors respectively. a_i and $m_i (i = 1,2,3,\ldots,n)$ are constant scalars and p × r constant matrices respectively. We introduce the shifting operator z defined by

$$y(k+i) = z^i y(k) \tag{2}$$

The equation (1) can be written as

$$(z^n + AZ_{1n})y(k) = M_{rn} Z_{rn} u(k) \tag{3}$$

where

$$A = (a_n \; a_{n-1} \; a_{n-2},\ldots,a_1) \qquad 1 \times n \tag{4}$$

$$M_{rn} = (m_n \; m_{n-1} \; m_{n-2},\ldots,m_1) \qquad p \times nr \tag{5}$$

$$Z_{1n} = (1 \; z \; z^2,\ldots,z^{n-1})^t \qquad n \times 1 \tag{6}$$

$$Z_{rn} = (1 \; z \; z^2,\ldots,z^{n-1})^t \otimes I_r \qquad nr \times r \tag{7}$$

t is a transposition, I_r is r × r identify matrix, ⊗ denotes the Kronecker product. It is convenient to indicate the relationship between the difference equation and the transfer function representation of the multi-variable systems. Applying Z-transformation on equation (1), assuming zero initial conditions, we can obtain the same formula as equation (3), so the transfer function matrix can be obtained

$$G(z) = (z^n + AZ_{1n})^{-1} M_{rn} Z_{rn} \tag{8}$$

Now, the problem is how to find the parametric A and M_{rn} using the I/O data u(k) and y(k). Let

$$Y_0^j = [y(k) \; y(k+1)\ldots y(k+j)] \tag{9}$$

$$U_0^j = [u(k) \; u(k+1)\ldots u(k+j)] \tag{10}$$

and using equation (3), we then have

$$Y_n^{n+j-1} = [-A\otimes I_p \; M_{rn}] \begin{bmatrix} Z_{pn} Y_0^{j-1} \\ Z_{rn} U_0^{j-1} \end{bmatrix} = [-A\otimes I_p \; M_{rn}] H_{(p+r)n} \tag{11}$$

where

$$H_{(p+r)n} = \begin{bmatrix} \\ Z_{pn} Y_0^{j-1} \\ \\ \\ Z_{rn} U_0^{j-1} \end{bmatrix} = \begin{bmatrix} y(k) & y(k+1)\ldots\ldots\ldots y(k+j-1) \\ y(k+1) & y(k+2)\ldots\ldots\ldots y(k+j) \\ \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ y(k+n-1) & y(k+n)\ldots\ldots\ldots y(\;\cdot\;) \\ u(k) & u(k+1)\ldots\ldots\ldots u(k+j-1) \\ u(k+1) & u(k+2)\ldots\ldots\ldots u(k+j) \\ \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ u(k+n-1) & u(k+n)\ldots\ldots\ldots u(\;\cdot\;) \end{bmatrix} \tag{12}$$

It is obvious, if $j = n(r+p)$, then the matrix $H_{(p+r)n}$ becomes a $n(r+p)\times n(r+p)$ square matrix. According to the Lemma given by Budin[3], if a system of (1) is completely observable and completely reachable, then for almost all input sequences

$$\rho[H_{(p+r)n}] = n+rn \neq n(r+p) \tag{13}$$

Therefore, $H_{(p+r)n}$ is singular, and the parametric matrix $[-A\otimes I_p \vdots M_{rn}]$ can't be obtained from (13) directly. But, if p=1, the input sequence is sufficiently random; that is, all vectors of U_0^j are linearly independent of row vectors of $H_{(p+r)n}$. Then $H_{(p+r)n}$ is nonsingular, $[A\otimes I_p \vdots M_{rn}]$ can be obtained easily. Basing on this reason many authors partition the system of (1) into p subsystems with multi-input and single-output. In this way, each subsystem can be identified independently and the identification of the whole system can be accomplished in p separate steps. If p is large, then there exists a large amount of unnecessary redundent work in computation. This idea, however, effects us to look for a characteristic subsystem with single-output and multi-input which has the same characteristic polynomial as the overall systems, so that we can obtain the parametric matrix at one step.

PARAMETRIC IDENTIFICATION

What subsystem can play a characteristic subsystem role? Theorem I will answer this question.

Theorem I: If the input sequence is persistently exciting and there exists a set of nonzero real number

$$C = (c_1\ c_2\ c_3 \ldots\ldots c_p) \tag{14}$$

that (z^n+AZ_{1n}) and $CM_{rn}Z_{rn}$ has no common factor, then the transfer function matrix can be computed at one step.

Proof: Suppose C is a set of nonzero real number. Multiplying both sides of (3) by C, we have

$$(z^n+AZ_{1n})Cy(k) = CM_{rn}Z_{rn}u(k) \tag{15}$$

Let $s(k) = Cy(k) \quad M = CM_{rn}$ (16)

The equation (15) can be written as

$$s(k+n) = [\ -A \vdots M\ ] \begin{bmatrix} Z_{1n}s(k) \\ Z_{rn}u(k) \end{bmatrix} \tag{17}$$

Let $S_i^j = [s(k+i)\ s(k+i+1) \ldots\ldots s(k+j)]$ (18)

then we have

$$\left[S_n^{2n-1} \vdots S_{2n}^{2n+nr-1} \right] = \left[-A \vdots M \right] \begin{bmatrix} Z_{1n} S_0^{n-1} & A_{1n} S_n^{n+nr-1} \\ Z_{rn} U_0^{n-1} & Z_{1n} U_n^{n+nr-1} \end{bmatrix} = \left[-A \vdots M \right] H_n \tag{19}$$

Because of the assumption of persistently exciting input sequence and the irreducible, Hankel matrix H_n is nonsingular and an unique solution for [-A M] exists. In addition, if $Z_{rn} U_n^{n+nr-1}$ is nonsingular, then the matrix A can be computed

$$A = - \left[S_n^{2n-1} - S_{2n}^{2n+nr-1} (Z_{rn} U_n^{n+nr-1})^{-1} (Z_{rn} U_0^{n-1}) \right] \left[Z_{1n} S_0^{n-1} - (Z_{1n} S_n^{n+nr-1}) (Z_{rn} U_n^{n+nr-1})^{-1} (Z_{rn} U_0^{n-1}) \right]^{-1} \tag{20}$$

From (11), we have

$$Y_n^{n+j-1} = - A \otimes I_p Z_{pn} Y_0^{j-1} + M_{rn} Z_{rn} U_0^{j-1} \tag{21}$$

Multiplying both sides of (21) by z^n, according to the property of the shifting operator

$$z^n U_0^j = U_n^{n+j} \qquad z^n Y_0^j = Y_n^{n+j} \tag{22}$$

and letting j = nr, (22) becomes

$$[\, A \otimes I_p \vdots I_p \,] Z_{p(n+1)} Y_n^{n+nr-1} = M_{rn} (Z_{rn} U_n^{n+nr-1}) \tag{23}$$

Note that the inverse of $(Z_{rn} U_n^{n+nr-1})$ has been calculated in processing A, so the matrix M_{rn} can be obtained easily

$$M_{rn} = [\, A \otimes I_p \vdots I_p \,] (Z_{p(n+1)} Y_n^{n+nr-1}) (Z_{rn} U_n^{n+nr-1})^{-1} \tag{24}$$

This completes the proof of the theorem.

DETERMINATION OF THE CHARACTERISTIC SYSTEMS

The algorithm discussed so far assumed that the characteristic subsystem has been found, and that $(z^n + AZ_{1n})$ and MZ_{rn} is irreducible. In practice, however, the characteristic subsystem is yet to be found, $(z^n + AZ_{1n})$ and MZ_{rn} will produce possibly a reducible part in the procedure of constructing the characteristic subsystem, although this possibility is very minute for a randomly chosen C from the real number space. (In the worst case C will be chosen for p times.)

We propose an algorithm for constructing the characteristic subsystem: the procedure is as follows:

Step 1: Construct the matrix $H_{(p+r)m}$ using the I/O sequences; determine its rank $\rho[H_{(p+r)m}]$ by the row searching algorithm [10]. The degree of the characteristic polynomial is given by

$$n = \rho[H_{(p+r)m}] - rm \qquad (25)$$

Step 2: Choose C from the real number space. It is the most convenient to choose $c_i=1$ $(i=1,2,3,\ldots,p)$. Construct H_m using S(k) and U(k); determine the order of this sybsystem m using Theorem II.

Step 3: Compare n with m; if n=m, that shows this subsystem is the characteristic subsystem from its definition. If not, repeat the procedure of step 2 until n=m. Theorem II is introduced not only for determining n but also for preparing the condition for computing the parametric matrices.

Theorem II: The order of a subsystem with single-output and multi-input is m if and only if the input sequence is persistently exciting and

$$\rho[\, R_m - D_m H_m^{-1} F_m \,] = r \qquad (26)$$

where

$$H_m = \begin{bmatrix} Z_{1m} S^{m+mr-1} \\ Z_{rn} U_0^{m+mr-1} \end{bmatrix} \qquad R_m = \begin{bmatrix} S_{m+mr}^{m+r(m+1)} \\ u_{n+mr}^{m+r(m+1)} \end{bmatrix} \qquad (27)$$

$$D_m = \begin{bmatrix} S_m^{m+mr} \\ u^{m+mr} \end{bmatrix} \qquad F_m = \begin{bmatrix} Z_{1m} S_{m+mr}^{m+r(m+1)} \\ Z_{rn} U_{m+mr}^{m+r(m+1)} \end{bmatrix} \qquad (28)$$

Proof:

Sufficiency: if $\rho[\, R_m - D_m H^{-1} F_m \,] = r$, suppose $n=m+1\neq m$. We make a new Hankel matrix H_{m+1} by adding the block rows consisting of the input and output sequences

$$H_{m+1} = \begin{bmatrix} H_m & F_m \\ D_m & R_m \end{bmatrix} \qquad (m+1)(r+1) \times (m+1)(r+1) \qquad (29)$$

Because any matrix doesn't change its rank by multiplying a nonsingular matrix,

$$\rho[H_{m+1}] = \rho[H_{m+1} Q_{m+1}] = \rho[H_m] + \rho[R_m - D_m H_m^{-1} f_m] \qquad (30)$$

where

$$Q_{m+1} = \begin{bmatrix} I & -H_m^{-1} F_m \\ 0 & I \end{bmatrix} \qquad (31)$$

For the persistently exciting sequence [11],

$$\rho[H_m] = \begin{cases} m+mr & m \leqslant n \\ n+mr & m > n \end{cases} \qquad (32)$$

(30) becomes

$$\rho[R_m - DH_m^{-1} F_m] = \rho[H_{m+1}] - \rho[H_m] = \gamma + 1 \qquad (33)$$

It contradicts the assumption.

Necessiry: if n=m, according to (30) and (32), then

$$\rho[\, R_m - D_m H_m^{-1} F_m \,] = \rho[\, H_{m+1} \,] - \rho\,[\, H_m \,] = r$$

EXAMPLE

The first example is a 2 × 2 two dimension system with the following transfer function matrix

$$G(z) = (z^2+2z+1)^{-1} \begin{bmatrix} 3(z+1) & z+1 \\ z-1 & 2z+1 \end{bmatrix}$$

For a set of input-output sequences of this system, see the following table

k	0	1	2	3	4	5	6	7	8
$u_1(k)$	1	0	0	1	0	0	1	0	0
$u_2(k)$	0	1	1	-1	-1	1	1	-1	-1
$y_1(k)$	0	3	-2	3	-1	0	1	3	-4
$y_2(k)$	0	1	-1	4	-7	6	-4	6	-10
$s(k)$	0	4	-3	7	-8	6	-3	9	-14

using the algorithm given in section (IV), starting from m = 1, k = 0, we can obtain n = 2.

$$H_2 = \begin{bmatrix} Z_{12}S_0^1 & Z_{12}S_2^5 \\ & \\ Z_{22}U_0^1 & Z_{22}U_2^5 \end{bmatrix} = \begin{bmatrix} 0 & 4 & -3 & 7 & -8 & 6 \\ 4 & -3 & 7 & -8 & 6 & -3 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & 1 \end{bmatrix}$$

$$(Z_{22}U_2^5)^{-1} = \begin{bmatrix} -1 & -1/2 & 1 & -1/2 \\ 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 0 \\ 1 & 1/2 & 0 & 1/2 \end{bmatrix}$$

$$S_2^3 - S_4^5(Z_{22}U_2^5)^{-1} = [\ \tfrac{-75}{2} \quad \tfrac{-26}{2}\ ]$$

$$Z_{12}S_0^1 - [(Z_{12}S_2^5)(Z_{22}U_2^5)^{-1}(Z_{22}U_0^1)]^{-1} = \begin{bmatrix} \frac{26}{117} & \frac{26}{117} \\ \frac{-66}{117} & \frac{-57}{117} \end{bmatrix}$$

Substituting these expressions into (20) yields

$$A = [\ \tfrac{-75}{2} \quad \tfrac{-26}{2}\ ] \begin{bmatrix} \frac{26}{117} & \frac{26}{117} \\ \frac{-66}{117} & \frac{-57}{117} \end{bmatrix} = [\ 1 \quad 2\ ]$$

Substituting A into the left side of (23) yields

$$[\, A \otimes I_p \vdots I_p \,]\,[\, Z_{23} Y_2^5 \,] = \begin{bmatrix} 1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} -2 & 3 & -1 & 0 \\ -1 & 4 & -7 & 6 \\ 3 & -1 & 0 & 1 \\ 4 & -7 & 6 & -4 \\ -1 & 0 & 1 & 3 \\ -7 & 6 & -4 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 3 & 1 & 0 & 5 \\ 0 & -4 & 1 & 4 \end{bmatrix}$$

From (24), we have

$$M_{22} = [\, A \otimes I_p \vdots I_p \,]\,(A_{23} Y_2^5)\,(Z_{22} U_2^5)^{-1} = \begin{bmatrix} 3 & 1 & 3 & 1 \\ 1 & -1 & 1 & 2 \end{bmatrix}$$

In the above example, with any nonzero real number C, we can identify the transfer function matrix at one step. But for the following example

$$G(z) = \begin{bmatrix} \frac{1}{z + 1/2} + \frac{3}{z + 1} & \frac{2}{z + 1/4} \\ \frac{-1}{z + 1/2} & \frac{5}{z + 1} \end{bmatrix}$$

if we still choose $c_i = 1$ $(i=1,2)$ and use the algorithm above, we will obtain $m = 2 \neq 3$, which is not true, because $\frac{1}{z+1/2}$ and $\frac{-1}{z+1/2}$ is cancelled in the procedure of constructing the characteristic subsystem. In this case, we will do it again with other C.

From this example, we see that only in the case where there exists common terms among the rows of a certain column of the transfer function matrix G(z) and provided that these terms do not exist in any other column , it is possible to compute more than once.

CONCLUSION

This paper gives an algorithm to identify a transfer function matrix from I/O measurements. When the input sequence is persistently exciting, the degree of the characteristic subsystem can be computed by a recursive algorithm in which one of the important by-products is that all conditions for computing the parametric matrices have already been prepared. The transfer function matrix can be obtained at one step, the computation is possibly simpler than other existing algorithms. The disadvantage of this algorithm is that the amount of work for choosing n has some uncertainty in particular cases, but even in the worst case, the computation is the same as [7] and [8].

ACKNOWLEDGMENT

The author wishes to thank Professor C.T. Chen for advice and encouragement.

REFERENCES

1. B.L. Ho and R.E. Kalman, "Eddective Construction of Linear State-Variable Model from Input-Output Function," Regelungstechnik 14, pp. 545-548, (1966).

2. B. Gopinath, "On the Identification of Linear Time-Invariant Systems for Input-Output Data," The Bell System Technical Journal, Vol. 48 No. 5, pp. 1101-1113, (1969).

3. M.A. Budin, "Minimal Realization of Discrete Linear Systems from Input-Output Observations," IEEE Trans. Auto. Control, Vol. AC-16, No. 5, pp. 395-401, (1971).

4. L.M. Siverman, "Realization of Linear Dynamical Systems," IEEE Trans. Auto. Control, AC-16, pp. 554-567, (1971).

5. H.H. Rosenbrok, "Computation of Minimal Representations of a Rational Transfer Function Matrix," Proc. IEEE 115, pp, 325-327, (1968).

6. J. Rissanen, "Recursive Identification of Linear Systems," SIAM J. Control 9, pp. 420-430, (1971).

7. D.P. Mital and C.T. Chen, "Identification of Multi-variable Systems from Input-Output Data," Proc. 2nd Cybernetics and Systems Conf., 1974.

8. Y.C. Wu and Z.V. Rekasius, "Identification of Multi-input Multi-output Linear Dynamic Systems," JACC Vol. 4, pp. 229-239, (1978).

9. N.K. Shinha and Y.H. Kwong, "Recursive Identification of the Parameters of Multi-variable System," Proc. 4th IFAC Symposium, pp. 323-327, (1977).

10. C.T. Chen, "Minimization of Linear Sequential Machines," IEEE Trans. Comput. Vol. C-23, pp. 93-95, (1974).

11. Y.C. Wu and Z.V. Rekasius, "Deterministic Identification of Linear Dynamic Systems," IEEE Trans. Auto. Control. Vol. AC-25, No. 3, pp. 501-504, (1980).

ON-LINE PARTIAL REALIZATIONS OF A DISCRETE-TIME NON-LINEAR BLACK BOX BY LINEAR REPRESENTATION SYSTEMS

TSUYOSHI MATSUO
Automatic Control Laboratory
Nagoya University
Nagoya 464 Japan

ABSTRACT

The problem of determining the internal state equations (linear representation system) of an unknown on-off causal black box (e.g., a controlled on-off process by computer) by on-line data is considered. One of our main theorems is that if the black box is described by a finite dimensional canonical linear representation system σ such that the next-state transitions are bijective, then there exists an input sequence ω such that σ can be determined up to isomorphisms by the output data of ω.

1. INTRODUCTION

More than 70% of control systems in the world may be on-off control systems. In spite of this fact, almost no attention is paid on on-off control systems by control theorists in these more than 15 years. Especially, almost no dynamical system descriptions which fit with on-off control systems exist.

In this paper, we introduce linear representation systems and show that this type of systems is adequate for the description (realization) of on-off controlled plants from experimental (input/output) data. By this fact we may design intelligent on-off controller.

We treat here discrete-time controlled plants with ON-OFF inputs and single output terminal, i.e., $T = \mathbb{N}$, $U = \{ON,OFF\}$ and $Y = K$, where K is a field. We think a controlled plant is a black box, i.e., we do not have any knowledge about the inside of the plant apriori. Then, we try to describe the inside of the plant from experimental (input/output) data. This has been called roughly identification or realization of the plant.

This paper treats the following three kinds of realization problems and solves them.

A. (Ideal) realization problem.

Given an input response map (ideal data of all allowable experiments), find a canonical linear representation system such that its input response map is the given one. And show that if there exists another such canonical system, then they are isomorphic.

B. Complete partial realization problem.

Given a complete partial input response map (finite data of all allowable experiments within N length), find a natural partial realization of it, and show that if there exists another natural partial realization, then they are isomorphic.

C. On-line partial realization problem.

For an input $\omega \in \mathrm{Mo}(U)$, let $a_\omega = \{(\omega(t),\gamma(t)) ; 0 \leq t \leq |\omega|\}$ be the input/output pairs of a black box. Find a minimum dimensional linear representation system which fits with the data. Show conditions when minimum dimensional systems are unique up to isomorphisms.

Problem A can be solved as an application of automata theory. See e.g., Eilenberg [1974], FLIESS [1974], SONTAG [1979b], and HASEGAWA and MATSUO [1979]. Problem B may be considered as a natural extension of the partial realization problem of linear systems. See KALMAN [1969]. Problem C may be our original. Recently, GRASELLI and ISIDORI [1977], SONTAG [1979a], and SUSSMANN [1979] discussed determination of an initial state by a single experiment. However, our problem is to determine a system itself, i.e., transition matrices, an initial state and a readout map, by a single experiment. MOORE [1956] had a result that if a finite suquential machine (in our language, a finite dynamical system without an initial state) is strongly connected (reachable and controllable) and distinguishable, then it is essentially determinable by a single experiment. Our theorem says that if a finite-dimensional linear representation system is quasi-reachable, reversible and distinguishable, then it is essentially determinable by a single experiment.

2. PRELININARIES

2.1 Input Set

For $U = \{\mathrm{ON},\mathrm{OFF}\}$, let $\mathrm{Mo}(U)$ be the monoid of words constructed by U, i.e., $\mathrm{Mo}(U) = \{u^n \cdots u^1 ; u^i \in U \text{ for } 1 \leq i \leq n,\ n \in \underline{N}\}$ and $\mathrm{Mo}(U) \times \mathrm{Mo}(U) \to \mathrm{Mo}(U) ; (u^n \cdots u^1, v^m \cdots v^1) \mapsto u^n \cdots u^1 v^m \cdots v^1$, with the conventin that the null word 1 belongs to $\mathrm{Mo}(U)$. A function $|\ | : \mathrm{Mo}(U) \to \underline{N} ; u^n \cdots u^1 \mapsto n$ is called the *length function*, where $\underline{N}$ is the set of non-negative integers. For any $m \in \underline{N}$, we consider the cutting $C(m) : \mathrm{Mo}(U) \to \mathrm{Mo}(U) ; u^n \cdots u^1 \mapsto u^k \cdots u^1$, where $k = \min(m,n)$.

We think that any element of $\mathrm{Mo}(U)$ is applicable to black boxes and systems. Note that $\mathrm{Mo}(U)$ is a free monoid generated by U and that $\mathrm{Mo}(U)$ satisfies the axioms of concatenation monoids (see MATSUO [1981, Definition (B,2.7)).

2.2 Input Response Maps

A map $a : \mathrm{Mo}(U) \to Y$ is called an *input response map*. It presents the complete experimental data of a black box by the relation

$$\gamma(|\omega|) = a(\omega) \quad \text{for all} \quad \omega \in \mathrm{Mo}(U)$$

where $\gamma(t)$ is the output at time $t \in \underline{N}$. Note the following :

(2.1) FACT (MATSUO [1981, Theorem (B,3.2))

Any causal input/output relation of a black box is uniquely described by an input response map, and vice versa.

2.3 Linear Representation Systems

Let X be a K-linear space, $F : U \to L(X)$, $x \mapsto X$ and $h \in L(X,Y)$, where $L(X,Y)$ is the set of all linear maps from X to Y and $L(X)$ is the set of all linear maps from X to X. A pair (X,F) is called a *linear U-action*, $((X,F),x^\circ)$ is called a *pointed linear U-action*, $((X,F),h)$ is called a *linear U-action with readout map*, and $\sigma = ((X,F),x^\circ,h)$ is called a *linear representation system.* The system equations of σ is given by

$$\begin{cases} x(0) = x^\circ \\ x(t+1) = F(\omega(t+1))x(t) \\ \gamma(t) = hx(t) \end{cases}$$

for all $t \in \underline{N}$ and $\omega(t) \in U$, where $x(t) \in X$ and $\gamma(t) \in Y$.

Consider the monoid morphism $\phi_F : Mo(U) \to L(X)$; $u^n \cdots u^1 \mapsto F(u^n) \cdots F(u^1)$ which is the unique extension of the map $F : U \to L(X)$. The *behavior* of σ is defined to be the input response map $a_\sigma : Mo(U) \to Y$; $\omega \mapsto h\phi_F(\omega)x^\circ$. If the linear hull of the reachable set $R(x^\circ) = \{\phi_F(\omega)x^\circ \ ; \ \omega \in Mo(U)\}$ is the whole space X, then $((X,F),x^\circ)$ [and σ] is called *quasi-reachable.* If $h\phi_F(\omega)x = 0$ for all $\omega \in Mo(U)$ imply $x = 0$, then $((X,F),h)$ [and σ] is called *distinguishable.* If σ is quasi-reachable and distinguishable, then it is called *canonical.*

(2.2) THEOREM. (HASEGAWA and MATSUO [1979])

For any input response map $a : Mo(U) \to Y$, *there exist at least two canonical linear representation systems with the behavior* a. (These systems are called realization of a).

Let $\sigma_1 = ((X_1,F_1),x_1^\circ,h_1)$ and $\sigma_2 = ((X_2,F_2),x_2^\circ,h_2)$ be linear representation systems. A linear map $f : X_1 \to X_2$ is called a (*linear*) *U-morphism* $f : (X_1,F_1) \to (X_2,F_2)$ if $fF_1(u) = F_2(u)f$ holds for all $u \in U$. Moreover, if $fx_1^\circ = x_2^\circ$ holds then it is called a *pointed U-morphism* $f : ((X_1,F_1),x_1^\circ) \to ((X_2,F_2),x_2^\circ)$, and if $h_2 f = h_1$ holds then it is called a *U-morphism with readout map* $f : ((X_1,F_1),h_1) \to ((X_2,F_2),h_2)$. If f is a pointed linear U-morphism with readout map, then it is called a *linear representation system morphism* $f : \sigma_1 \to \sigma_2$. Moreover, if f is bijective, then $f : \sigma_1 \to \sigma_2$ is called an *isomorphism.*

(2.3) THEOREM. (HASEGAWA and MATSUO [1979])

Two canonical linear representation systems σ_1 *and* σ_2 *are isomorphic, i.e., there exists an isomorphism* $f : \sigma_1 \to \sigma_2$ (*it is eventually unique*) *if and only if their behaviors are same.*

3. LINEAR U-ACTIONS

Here, we present examples of linear U-actions, add some terminologies, and give formulas of linear U-morphisms.

(3.1) EXAMPLE. Let K<U> be the K-linear space of all polynomials of noncommutative variable $u \in U$, i.e., $K\langle U\rangle = \{\sum_{\omega}\lambda(\omega)\omega \; ; \; \omega \in Mo(U)$ and $\lambda(\omega) \in K\}$. For each $u \in U$, let
$S_r(u) : K\langle U\rangle \to K\langle U\rangle \; ; \; \sum_{\omega}\lambda(\omega)\omega \mapsto \sum_{\omega}\lambda(\omega)(u\omega)$.
Then $(K\langle U\rangle, S_r)$ is a linear U-action. We denote also S_r the monoid morphism which is the extension of S_r, i.e.,
$S_r : Mo(U) \to L(K\langle U\rangle) \; ; \; \omega \mapsto [\sum_{\omega'} \lambda(\omega)\omega' \mapsto \sum_{\omega'} \lambda(\omega')(\omega\omega')]$.

(3.2) EXAMPLE. Consider the linear space F(Mo(U),Y) of all input response maps. For each $u \in U$, let
$S_\ell(u) : F(Mo(U),Y) \to F(Mo(U),Y) \; ; \; a \mapsto [\omega \mapsto a(\omega\ u)]$. Then $(F(Mo(U),Y),S_\ell)$ is a linear U-action. We denote also S_ℓ the monoid morphism which is the extension of S_ℓ.

REMARK. The set K<U> of all polynomials of noncommutative variable is a noncommutative K-algebra by the usual polynomial multiplication. See FLIESS [1974]. Then, a linear U-action (X,F) is considered as a K<U>-module by the action $K\langle U\rangle \times X \to X \; ; \; (\sum_{\omega}\lambda(\omega)\omega, x) \mapsto \sum_{\omega}\lambda(\omega)\phi_F(\omega)x$. Here, linear U-morphisms are K<U>-morphisms.

Let (X,F) be a linear U-action. A linear subset $S \subseteq X$ is called *invariant under* F if $F(u)S \subseteq S$ for all $u \in U$. Let W be a subset of X. The minimum invariant subspace of X which contains W (i.e., the intersection of all invariant subspaces of X which contain W) is called the *module hull* of W and denoted by [W]. Let $W = \{x_\iota \in X \; ; \; \iota \in I\}$ where I is an index set. Then W is called a *generator* (X,F) if [W] = X, and W is called *free* if $[x_\iota] \cap [x_\kappa] = \{0\}$ for any $\iota \neq \kappa \in I$. A linear U-action (X,F) is called *finitely generated* if there exists a finite generator and it is called *monogenic* if there exists a singleton generator. A linear U-action (X,F) is called *simple* if there exist no nontrivial invariant subspaces, i.e., there exist no invariant subspaces except X and {0}. Hence, (X,F) is simple if and only if {x}, $0 \neq x \in X$ is a generator of X. A linear U-action (X,F) is called *reversible* if F(u), $u \in U$ are bijective.

(3.3) LEMMA. *A linear U-morphism* $G : (K\langle U\rangle, S_r) \to (X,F)$ *is given by* $G : \sum_{\omega}\lambda(\omega)\omega \mapsto \sum_{\omega}\lambda(\omega)\phi_F(\omega)x^\circ$, *where* $x^\circ = G(1)$. *A linear U-morphism* $H : (X,F) \to (F(Mo(U),Y),S_\ell)$ *is given by* $H : x \mapsto [\omega \mapsto h\phi_F(\omega)x]$, *where* $h : X \to Y \; ; \; x \mapsto (Hx)(1)$.

In the category of linear U-actions, we have subspaces, quotient subspaces, product spaces, spaces of direct sum, and dual spaces. Also, linear U-morphisms admit canonical decomposition.

4. VARIOUS QUASI-REACHABILITIES

(4.1) PROPOSITION. *In the category of pointed linear* U-*actions,* $((K\langle U\rangle, S_r), 1)$ *is an initial object, i.e., for any pointed linear* U-*action* $((X,F),x^\circ)$, *there exists a uniquely determined pointed linear* U-*morphism* $G : ((K\langle U\rangle, S_r), 1) \to ((X,F),x^\circ)$ *which is given by*

$G : \sum_{\omega}\lambda(\omega)\omega \mapsto \sum_{\omega}\lambda(\omega)\phi_F(\omega)x^\circ$.

PROOF. By Lemma (3.3), we have the proposition.

The morphism G in Proposition (4.1) is called the *linear input map* of $((X,F),x^\circ)$.

(4.2) EXAMPLE. Consider the pointed linear U-action $((F(Mo(U),Y),S_\ell),a)$ for an input response map a. Then the linear input map A of it is given by

$A : K\langle U\rangle \to F(Mo(U),Y) \; ; \; \sum\lambda(\omega)\omega \mapsto [\omega' \mapsto \sum\lambda(\omega)a(\omega'\omega)]$.

The map is called the *linear input/output map* of a. By taking an appropriate linear basis, A is represented by the matrix

$$H_a = \begin{array}{cc} \omega & \\ \left[\begin{array}{c} \vdots \\ a(\omega'\omega) \;\cdots \\ \\ \end{array}\right] & \omega' \end{array} \qquad \text{where } \omega, \omega' \in K\langle U\rangle$$

which is called the *Hankel matrix* of a by FLIESS [1974].

(4.3) PROPOSITION. *A pointed linear* U-*action* $((X,F),x^\circ)$ *is quasi-reachable if and only if the linear input map* G *is surjective. Hence, if* $((X,F),x^\circ)$ *is quasi-reachable, then it is isomorphic to* $((K\langle U\rangle/\ker G, \hat{S}_r), 1+\ker G)$, *where* $\hat{S}_r$ *is the induced transition map from* S_r. *Therefore, the isomorphic class of quasi-reachable* $((X,F),x^\circ)$ *is characterized by the invariant subspace* ker G *of* $(K\langle U\rangle, S_r)$.

PROOF. It is the direct consequence of Proposition (4.1) and the basic properties of linear U-actions.

(4.4) COROLLARY. *A quasi-reachable pointed linear* U-*action* $((X,F),x^\circ)$ *is reversible if and only if the following two conditions hold.*

a) *For any* $u \in U$, $S_r(u)\lambda \in \ker G$ *implies* $\lambda \in \ker G$.

b) *For any* $u \in U$ *and* $\lambda \in K\langle U\rangle$, *there exists* $\lambda' \in K\langle U\rangle$ *such that* $\lambda - S_r(u)\lambda' \in \ker G$.

Consider a pointed linear U-action $((X,F),x^\circ)$ and a positive integer n. The linear hull $QR(x^\circ, \leq n)$ of the set $\{\phi_F(\omega)x^\circ \in X \; ; \; |\omega| \leq n\}$ is called the *quasi-reachable subspace within length* n. If $QR(x^\circ, \leq n) = X$

it is called *quasi-reachable within length* n. The quasi-reachable space within length n of $((K\langle U\rangle, S_r),1)$ is $K\langle U,\le n\rangle = \{\sum_{|\omega|\le n} \lambda(\omega)\omega \in K\langle U\rangle\}$. Let $G(\le n)$ be the restriction to $K\langle U,\le n\rangle$ of the linear input map. Then $((X,F),x^\circ)$ is quasi-reachable within length n if and only if $G(\le n)$ is surjective. We have the following proposition.

(4.5) PROPOSITION. *Let* n *be positive integer.* A n-*dimensional quasi-reachable pointed linear* U-*action is quasi-reachable within length* $n-1$.

(4.6) LEMMA. *Let* n *be a positive integer and let* S_n *be a linear subspace of* $K\langle U,\le n\rangle$ *such that the following conditions hold.*

a) *If* $\lambda \in K\langle U,\le n-1\rangle \cap S_n$, *then* $S_r(u)\lambda \in S_n$ *for all* $u \in U$.

b) *For any* $\lambda \in K\langle U,\le n\rangle$, *there exists a* $\lambda' \in K\langle U,\le n-1\rangle$ *such that* $\lambda-\lambda' \in S_n$

Then there exists a uniquely determined invariant subspace S *of* $(K\langle U\rangle, S_r)$ *such that* $S \cap K\langle U,\le n\rangle = S_n$ *and* $K\langle U\rangle/S$ *is isomorphic to* $K\langle U,\le n\rangle/S_n$. *Moreover, the pointed linear* U-*action* $((K\langle U\rangle/S, \hat{S}_r), 1+S)$ *is quasi-reachable within length* $n-1$.

PROOF. See MATSUO [1981, Corollary (C,4.26)].

A pointed linear U-action $((X,F),x^\circ)$ is called *quasi-reachable* by $\omega \in Mo(U)$ if the linear hull of the set $\{\phi_F(C(i)\omega)x^\circ \ ; \ 0 \le i \le |\omega|\}$ is the whole space X.

(4.7) LEMMA. *Let* $((X,F),x^\circ)$ *be a pointed linear* U-*action and consider the following conditions. Then the implication* a)$\Rightarrow$b)$\Rightarrow$c)$\Rightarrow$d) *holds.*

a) (X,F) *is simple.*

b) $((X,F),x^\circ)$ *is reversible and quasi-reachable.*

c) *For any* $x \in R(x^\circ) = \{\phi_F(\omega)x^\circ \ ; \ \omega \in Mo(U)\}$, $[x] = X$.

d) *For any* $\omega \in Mo(U)$, *there exists an* $\omega' \in Mo(U)$ *such that* $((X,F),x^\circ)$ *is quasi-reachable by* $\omega'\omega$.

PROOF. a)$\Rightarrow$b). Obvious by definition. b)$\Rightarrow$c). Assume that there exists $x' \in R(x^\circ)$ such that $[x'] \subsetneq X$. There exist $x_1 \in [x']$, $x_2 \in X\backslash[x']$ and $u \in U$ such that $F(u)x_2 = x_1$. Since $F(u)$ is bijective and $[x']$ is invariant under $F(u)$, there exists $x_3 \in [x']$ such that $F(u)x_3 = x_1$. This contradicts with the bijection of $F(u)$.
c)$\Rightarrow$d). Let $\omega \in Mo(U)$, $S = \{\phi_F(C(i)\omega)x^\circ \ ; \ 0 \le i \le |\omega|\}$, and $[S] \subsetneq X$. Since $((X,F),x^\circ)$ is quasi-reachable, there exists an $\omega' \in Mo(U)$ such that $\phi_F(\omega'\omega)x^\circ \notin [S]$.

5. CONSTRUCTIBILITY AND OBSERVABILITY

(5.1) PROPOSITION. *In the category of linear* U-*actions with readout* *maps,* $((F(Mo(U),Y),S_\ell),1)$ *where* $1 : a \mapsto a(1)$ *is a final object, i.e., for any linear* U*-action with readout map* $((X,F),h)$*, there exists a uniquely determined linear* U*-morphism with readout map* $H : ((X,F),h) \to ((F(Mo(U),Y),S_\ell),1)$ *which is given by*

$H : x \mapsto [\omega \mapsto h\phi_F(\omega)x]$.

This proposition is a direct consequence of Lemma (3.3). The morphism H in the proposition is called the *observation map* of $((X,F),h)$.

(5.2) PROPOSITION. *A linear* U*-action with readout map* $((X,F),h)$ *is distinguishable if and only if the observation map* H *is injective. Hence, if* $((X,F),h)$ *is distinguishable, then it is isomorphic to* $((\text{im } H,S_\ell),1)$.

PROOF. It is the direct consequence of Proposition (5.1) and the basic properties of linear U-actions.

(5.3) LEMMA. Let $\sigma_1 = ((X_1,F_1),\mathring{x}_1,h_1)$ and $\sigma_2 = ((X_2,F_2),\mathring{x}_2,h_2)$ be canonical linear representation systems such that the condition c) of Lemma (4.7) holds. Then $\text{im } H_1 = \text{im } H_2$ or $\text{im } H_1 \cap \text{im } H_2 = 0$.

REMARK. This can be proved by the same way as MATSUO [1981, Lemma (C,5.21)]. MOORE [1956] had the same type of lemma by using strongly connectedness.

Consider a linear U-action with readout map $((X,F),h)$ and a positive integer n. It is called *distinguishable within length* n if $h\phi_F(\omega)x = 0$ for all $\omega \in Mo(U, \leq n)$ imply $x = 0$. Let $F(Mo(U, \leq n),Y)$ be the set of all complete partial input response maps, and let $H(\leq n)$ be the restricted map of the observation map H to $F(Mo(U, \leq n),Y)$. Then, $((X,F),h)$ is distinguishable within n if and only if $H(\leq n)$ is injective. We have the following proposition.

(5.4) PROPOSITION. *Let* n *be a positive integer.* A n*-dimensional linear* U*-action with readout map* $((X,F),h)$ *is distinguishable within* n-1.

(5.5) LEMMA. Let n be a positive integer and Z_n be a linear subspace of $F(Mo(U, \leq n),Y)$ such that the following conditions hold.

a) The map $Z_n \xrightarrow{j} F(Mo(U, \leq n),Y) \xrightarrow{\pi} F(Mo(U, \leq n-1),Y)$ is injective, where j is the canonical injection and π is the natural projection.

b) For any $\omega \in Mo(U, \leq n)$ and $u \in U$, $\text{im } (\tilde{S}_\ell(u) \cdot j) \subseteq \text{im } (\pi \cdot j)$, where $\tilde{S}_\ell(u) : F(Mo(U, \leq n),Y) \to F(Mo(U, \leq n-1),Y)$

is the restricted map of $S_\ell(u)$.

Then, there exists a uniquely determined invariant subspace Z of $(F(Mo(U),Y),S_\ell)$ such that the natural projection $\pi : Z \to Z_n$ is bijective. Moreover, the linear U-action $((Z,S_\ell),1)$ is distinguishable within length n-1.

PROOF. See MATSUO [1981, Lemma (C,5.22)].

A linear U-action with readout map $((X,F),h)$ is called *observable by* $\omega \in Mo(U)$ if $h\phi_F(C(i)\omega)x = 0$ for all $0 \le i \le |\omega|$ imply $x = 0$. It is called *constructible* by ω if $h\phi_F(C(i)\omega)x = 0$ for all $0 \le i \le |\omega|$ imply $\phi_F(\omega)x = 0$. "Observable by ω" implies "constructible by ω", however, if (X,F) is reversible, the converse holds. First, we need the following lemma.

(5.6) LEMMA. *If* $((X,F),h)$ *is distinguishable, then* ker h *contains no invariant subspaces of* (X,F). *If* X *is finite-dimensional, the converse holds.*

PROOF. The first part is obvious by definition. The second part can be proved by the same argument as in the proof of Lemma (4.7).

(5.7) PROPOSITION. *If* $((K^n,F),h)$ *is distinguishable, then there exists an input* $\omega \in Mo(U, \le n(n-1)/2)$ *such that it is constructible by* ω.

PROOF. Unconstructible space by null input 1 is S_1 = ker h. Since S_1 is not invariant under F, there exists an input $\omega_1 = u \in U$ such that the dimension of the unconstructible space S_2 by ω_1 is less than dim S_1. Since S_2 is not invariant under F, there exists an input ω_2 such that $|\omega_2| \le 2$ and the dimension of the unconstructible space by $\omega_2\omega_1$ is less than dim S_2. Continuing this procedure, we have the result.

6. COMPLETE PARTIAL REALIZATIONS.

For an input response map $a : Mo(U) \to Y$, let $A : K\langle U\rangle \to F(Mo(U),Y)$ be its linear input/output map. Then

$\sigma_q = ((K\langle U\rangle/\ker A, S_r), 1+\ker A, \hat{a})$ where $a = \hat{a}\cdot\pi$ and

$\pi : K\langle U\rangle \to K\langle U\rangle/\ker A$ is the canonical surjection, and

$\sigma_s = ((\mathrm{im}\, A, S_\ell), a, 1)$ are canonical realization of a which are mentioned in Theorem (2.2). The input response map a admits n-dimensional canonical realizations if and only if im A is n-dimensional, i.e., the Hankel matrix H_a has rank n.

Now consider the situation where a complete partial input response map $a_N : Mo(U, \le N) \to Y$ is given. A linear representation system $\sigma = ((X,F),x°,h)$ is called a *partial realization* of a_N if $h\phi_F(\omega)x° = a_N(\omega)$ for all $\omega \in Mo(U, \le N)$. For any complete partial input response map, there exist finite-dimensional partial realizations, hence, there exists a minimum dimensional partial realization. It is canonical, however, it is not unique in general even up to isomorphisms.

A partial realization σ of a_N is called *natural* if σ is quasi-reachable within length n_1 and distinguishable within length n_2 where $n_1+n_2 < N$.

For a complete partial input response map $a_N : Mo(U, \leq N) \rightarrow Y$ and positive integers n_1, n_2 such that $0 \leq n_1+n_2 < N$, form the *partial linear input/output map*

$A(n_1,n_2) : K<U, \leq n_1> \longrightarrow F(Mo(U, \leq n_2),Y)$; $\sum_{\omega}\lambda(\omega)\omega \mapsto [\bar{\omega} \mapsto \sum_{\omega}\lambda(\omega)a(\bar{\omega}\omega)]$ or in the Hankel matrix form

$$H_a(n_1,n_2) = \begin{array}{cl} \omega & \\ \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \\ a(\bar{\omega}\omega)\ \cdot\ \cdot\ \cdot \\ \\ \end{array}\right] & \begin{array}{l} \\ \\ \\ \bar{\omega} \\ \\ \end{array} \end{array}, \qquad \begin{array}{l} \text{where } \omega \in Mo(U, \leq n_1) \\ \text{and } \bar{\omega} \in Mo(U, \leq n_2) \end{array}$$

(6.1) THEOREM. *A complete partial input response map* $a_N : Mo(U, \leq N) \rightarrow Y$ *admits a natural partial realization if and only if there exists a positive integer* $0 \leq n_1 < N$ *such that the following conditions hold.*

a) $\ker A(n_1,N-n_1-1) = \ker A(n_1,N-n_1)$

b) $\operatorname{im} A(n_1,N-n_1-1) = \operatorname{im} A(n_1+1,N-n_1-1)$

These conditions are equivalent to

c) $\operatorname{rank} H_a(n_1,N-n_1-1) = \operatorname{rank} H_a(n_1,N-n_1) = \operatorname{rank} H_a(n_1+1, N-n_1-1)$.

Moreover, if for all $u \in U$

d) $\operatorname{rank} H_a(n_1,N-n_1) = \operatorname{rank} S_\ell(u)H_a(n_1,N-n_1)$.

where $S_\ell(u)H_a(n_1,N-n_1)$ *is the matrix representation of* $S_\ell(u)A(n_1,N-n_1)$, *then natural partial realizations are reversible.*

Two natural partial realizations of a_N *are isomorphic.*

PROOF. By using the property of partial linear input/output maps and Lemmas (4.6) and (5.5), we can prove the theorem by the same way as MATSUO [1981, Theorem (B,8.14)]. Reversibility comes from Corollary (4.4) and Lemma (4.6).

7. ON-LINE PARTIAL REALIZATIONS

Given an input $\omega \in Mo(U)$, a map $a_\omega : [0,|\omega|] \rightarrow Y$ is called an *on-line partial input response map*. A linear representation system $\sigma = ((X,F),x^\circ,h)$ is called a *partial realization* of a_ω if $a_\omega(t) = h\phi_F(C(t)\omega)x^\circ$ holds for $0 \leq t \leq |\omega|$. For any on-line partial input response map, there exist finite dimensional partial realizations, hence, there exists a minimum dimensional partial realization. In general, however, it is not unique even up to isomorphisms. Uniqueness property is important, since when we calculate minimum dimensional partial realizations successively according to the application of new input values, the uniqueness garantees the convergence of the algorithm.

(7.1) THEOREM. *Assume that the unknown plant is described by a finite dimensional reversible linear representation system* σ *with the dimension no more than the known number* N. *Then there exists an input such that* σ *can be determined uniquely up to isomorphisms by the on-line partial input response map* a_ω.

SCKETCH OF PROOF. By Theorem (6.1), the assumptions of the theorem imply that the input response map of σ is determined by an element of the set $F(Mo(U, \leq 2N-1), Y)$ of all complete partial input response maps such that the conditions c) and d) of Theorem (6.1) hold. Let W be the linear hull of the set of all input response maps which might be the behavior of σ. Then $W \subset F(Mo(U),Y)$ is finite dimensional by the above arguments, W is an invariant subspace of $(F(Mo(U),Y),S_\ell)$, and $((W,S_\ell),1)$ is reversible distinguishable linear U-actions with readout map. Then, by Proposition (5.7) and reversibility, there exists an input $\omega \in Mo(U)$ such that $((W,S_\ell),1)$ is observable by ω. Hence we have a_σ.

REFERENCES

S. EILENBERG [1974]. *Automata, Languages and Machines,* Academic Press, New York.

M. FLIESS [1974]. Matrices de Hankel. J. Math. Pures et Appl. *53* : 197-222.

O. M. GRASSELLI and A. ISIDORI [1977]. Deterministic state reconstruction and reachability of linear control processes, *Proc. J.A.C.C.,* San Francisco, June 22-25.

Y. HASEGAWA and T. MATSUO [1979]. Realization theory of discrete-time linear representation systems. SICE Transactions, *15* : 298-305, (in Japanese).

R. E. KALMAN, P. L. FALB and M. A. ARBIB [1969]. *Topics in Mathematical system Theory,* McGraw-Hill, New York.

T. MATSUO [1981]. *Realization Theory of Continuous-Time Dynamical Systems,* Lecture Notes in Control and Information Sciences, Springer, Berlin.

E. F. MOORE [1956]. Gedanken-experiments on sequential machines, in *Automata Studies,* Princeton Univ. Press, Princeton, New Jersey.

E. SONTAG [1979a]. On the observability of polynomial systems, SIAM J. Control and Optimization *19* : 139-151.

――――― [1979b]. Realization theory of discrete time nonlinear systems : Part I, IEEE Transactions on Circuits and Systems, *CAS-26* : 342-356.

H. J. SUSSMANN [1979]. Single input observability of continuous-time systems, Math. System Theory *12* : 371-393.

IDENTIFICATION AND MODEL APPROXIMATION FOR CONTINUOUS-TIME SYSTEMS ON FINITE PARAMETER SETS

JITENDRA K. TUGNAIT
Department of Electrical and Computer Engineering
University of Iowa
Iowa City, Iowa 52242

ABSTRACT

Almost sure convergence of the maximum likelihood estimates of unknown parameters of continuous-time stochastic dynamical linear time-invariant systems is investigated. The unknown parameter set is assumed to be finite. The situation where the true parameter does not belong to the unknown parameter set is considered as well as the situation where the true model is included in the unknown parameter set.

I. INTRODUCTION

The problem of identifying parameters of stochastic systems has been studied extensively [7-13], [16-23]; however, only a few [1-6] have treated the continuous-time systems. In this note we give sufficient conditions for strong consistency of the maximum likelihood (ML) parameter estimates on finite sets for continuous linear time-invariant systems. While the results in [1,3,5] were patterned on the work of Crámer [14], our results follow Wald [15]. The approach of this note allows convergence analysis of the case where the true parameter does not belong to the unknown parameter set, unlike the approach of [1,3,5].

Consider the following continuous-time stochastic linear dynamical system

$$x(t;\omega) = \int_0^t Ax(s;\omega)ds + \int_0^t F\, dW_1(s;\omega) + x(0;\omega) \tag{1}$$

$$y(t;\omega) = \int_0^t Cx(s;\omega)ds + \int_0^t G\, dW_2(s;\omega) \tag{2}$$

where $x(t;\omega)$ and $y(t;\omega)$ are n- and m-dimensional state and observation vectors, respectively; A, C, F and G and $n \times n$, $m \times n$, $n \times n$, and $m \times m$ constant matrices, respectively; $W_1(t;\omega)$ and $W_2(t;\omega)$ are n- and m-dimensional independent Wiener processes, respectively, with identify covariances. We have $x(0;\omega) = 0$ and $W_1(0;\omega) = W_2(0;\omega) = 0$. Finally, we assume that $GG^* > 0$[†] to exclude the singular case.

The identification problem is that of estimating unknown parameters in the matrices A, C, and F, given $y(t;\omega)$, $0 \le t \le T$, for one realization of the experiment. We assume that G is known completely.

Let θ denote the vector of all the unknown system parameters and let θ_0 be the true value.[††] It is assumed that $\theta \in \Theta$, where Θ contains a finite number of elements. Let $\hat{\theta}_T(\omega)$ denote the ML estimate of θ on Θ based on the measurements $y(t;\omega)$ $0 \leq t \leq T$. An ML estimate on Θ is an estimate $\hat{\theta}_T(\omega) \in \Theta$ such that

---

[†] G^* denotes transpose of G, and $GG^* > 0$ denotes the matrix GG^* is positive definite.

[††] Unless otherwise indicated, the probability measure used in this paper is the one corresponding to the true parameter θ_0.

$$H(\hat{\theta}_T(\omega);y(\cdot;\omega)T) \geq H(\theta;y(\cdot,\omega);T) \quad (\forall\theta \in \Theta)$$

where $H(\theta;y(\cdot;\omega);T)$ is a likelihood functional given by [1, Theorem 8.1]:

$$H(\theta;y(\cdot;\omega);T) = \exp\left[-\frac{1}{2}\left(\int_0^T [(GG^*)^{-1}\, C_\theta\hat{x}(t;\theta;\omega),\, C_\theta\hat{x}(t;\theta;\omega)]dt \right.\right.$$
$$\left.\left. - 2\int_0^T [(GG^*)^{-1}\, C_\theta\hat{x}(t;\theta;\omega),\, dy(t;\omega)]\right)\right] \tag{3}$$

where $[\cdot,\cdot]$ denotes inner product in R^m and the second integral is to be interpreted in the Itô sense. Note that we indicate the dependence of various quantities in (3) on θ explicitly. Furthermore, $\hat{x}(t;\theta;\omega)$ satisfies the stochastic differential equation.

$$\hat{x}(t;\theta;\omega) = \int_0^t A_\theta\hat{x}(s;\theta;\omega)ds + \int_0^t P_\theta(s)C_\theta^*(GG^*)^{-1}dz(s;\theta;\omega) \tag{4}$$

$$z(t;\theta;\omega) = y(t;\omega) - \int_0^t C_\theta\hat{x}(s;\theta;\omega)ds \tag{5}$$

where $P_\theta(t)$ satisfies the matrix Riccati equation

$$dP_\theta(t)/dt = A_\theta P_\theta(t) + P_\theta(t)A_\theta^* + F_\theta F_\theta^* - P_\theta(t)C_\theta^*(GG^*)^{-1}C_\theta P_\theta(t);$$
$$P_\theta(0) = 0. \tag{6}$$

Here $\hat{x}(t;\theta;\omega) = E\{x(t;\omega)|\mathcal{B}_y(t),\theta\}$ where $\mathcal{B}_y(t)$ is the smallest σ-algebra generated by $y(\tau;\omega)$, $0 \leq \tau \leq t$. Furthermore, $P_\theta(t)$ is the corresponding state estimation error covariance, and $z(t;\theta;\omega)$ is a Wiener process, the so-called innovations process, for $\theta = \theta_0$, the true parameter value, with

$$E\{z(t;\theta_0;\omega)z^*(s;\theta_0;\omega)\} = (GG^*)\min(t,s). \tag{7}$$

Now let

$$q(\theta;y(\cdot;\omega);T) = -(2/T)\ell n\, H(\theta;y(\cdot;\omega);T). \tag{8}$$

Conditions (C1) - (C2) are imposed on the system (1) - (2) to ensure convergence of certain expressions as t becomes very large.

(C1) For any θ in $\Theta \cup \{\theta_0\}$, A_θ is a stable matrix, i.e., all the eigenvalues of A have negative real parts.

(C2) For any θ in $\Theta \cup \{\theta_0\}$, the pair (C_θ, A_θ) is completely observable, that is, rank $[C_\theta^* \mid A_\theta^*C_\theta^* \mid \ldots \mid (A_\theta^*)^{n-1}C_\theta^*] = n$.

Condition (C1) implies that the initial condition of the state has no effect on the asymptotic behavior of the system and this justifies setting the initial condition to be zero in (1). Condition (C2) implies that

$\lim_{T\to\infty} P_\theta(t)$ exists and is unique; we denote this limit by P_θ, which satisfies the algebraic equation

$$A_\theta P_\theta + P_\theta A^*_\theta + F_\theta F^*_\theta - P_\theta C^*_\theta (GG^*)^{-1} C_\theta P_\theta = 0 \tag{9}$$

and furthermore, the matrix $A_\theta - P_\theta C^*_\theta (GG^*)^{-1} C_\theta$ is stable.

Let $q(\theta;T) = E\{q(\theta;y(\cdot;\omega);T)\}$ and define $q(\theta) = \lim_{T\to\infty} q(\theta;T)$. In [1], it has been shown that $q(\theta)$ exists.

II. CONVERGENCE OF PARAMETER ESTIMATES

In this section we investigate sufficient conditions for a.s. convergence of the ML estimates of θ on Θ. We consider both the cases: when $\theta_0 \in \Theta$ and when $\theta_0 \notin \Theta$. We shall use stationary version of the likelihood functional which is obtained by using the steady-state version of the Kalman-Bucy filter (4) - (5) ($\forall\theta \in \Theta$) and by assuming that system (1) - (2) started in "remote" past.

From (1), (2), (4), and (5), we have

$$\begin{bmatrix} x(t,\omega) \\ \hat{x}(t;\theta;\omega) \end{bmatrix} = \int_0^t A' \begin{bmatrix} x(s;\omega) \\ \hat{x}(s;\theta;\omega) \end{bmatrix} ds + \int_0^t F' \begin{bmatrix} dW_1(s;\omega) \\ dW_2(s;\omega) \end{bmatrix} \tag{10}$$

where

$$A' \equiv \begin{bmatrix} A_{\theta_0} & 0 \\ P_\theta C^*_\theta (GG^*)^{-1} C_{\theta_0} & A_\theta - P_\theta C^*_\theta (GG^*)^{-1} C_\theta \end{bmatrix}$$

$$F' \equiv \begin{bmatrix} F_{\theta_0} & 0 \\ 0 & P_\theta C^*_\theta (GG^*)^{-1} G \end{bmatrix}$$

and where we have used steady-state version of the Kalman-Bucy filter (4) - (5) $\forall\theta \in \Theta$ so that P_θ is given by the positive semi-definite solution of (9); there is no loss of generality in this, however, algebraic manipulations are considerably reduced. Now let

$$L(t;\theta;\theta_0) \triangleq E\left\{ \begin{bmatrix} x(t;\omega) \\ \hat{x}(t;\theta;\omega) \end{bmatrix} [x^*(t;\omega)\ \hat{x}^*(t;\theta;\omega)] \right\} \tag{11}$$

Then, from (10) and (11), we have [1, Equation (2-13)]

$$dL(t;\theta;\theta_0)/dt = A'L(t;\theta;\theta_0) + L(t;\theta;\theta_0)A'^* + F'F'^* \tag{12}$$

Now $L(\theta;\theta_0) \triangleq \lim_{t\to\infty} L(t;\theta;\theta_0)$ exists (irrespective of initial condition) if A' is stable (i.e., if all eigenvalues of A' have negative real parts) [1, p. 144]; this is indeed true $\forall\theta \in \Theta$ because of conditions (C1) - (C2). Therefore, $L(\theta;\theta_0)$ satisfies the algebraic equation

$$A'L(\theta;\theta_0) + L(\theta;\theta_0)A'^* + F'F'^* = 0. \tag{13}$$

Define

$$\hat{x}_1(t;\theta;\omega) \triangleq C_\theta \hat{x}(t;\theta;\omega) \tag{14}$$

$$x_1(t;\omega) \triangleq C_{\theta_0} x(t;\omega) \tag{15}$$

$$P_1^0(t;\theta) \triangleq E\{[x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)][x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^*\}. \tag{16}$$

We note again that expectations in (16) is taken w.r.t. the true probability measure corresponding to θ_0. Therefore, $P_1^0(t;\theta)$ denotes the estimation error covariance in estimating $x_1(t)$ using a filter corresponding to θ when θ_0 is the correct parameter value. Finally

$$\lim_{t\to\infty} P_1^0(t,\theta) \triangleq P_1^0(\theta) = C'L(\theta;\theta_0)C'^* \tag{17}$$

where $C' \equiv [C_{\theta_0} - C_\theta]$.

Note that if we take initial time to be t_0 instead of zero, then the preceding discussion also implies that

$$P_1^0(\theta) = \lim_{t_0\to-\infty} E\{[x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)][x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^*\}. \tag{18}$$

Lemma 1. Suppose that conditions (C1) - (C2) are satisfied and that system (1) - (2) is started in remote past (i.e., initial time $t_0 = -\infty$ instead of $t_0 = 0$). Then the process $(x_1(t;\omega) - \hat{x}_1(t;\theta;\omega))$ is ergodic for $t \geq 0$ and for every θ in $\Theta \cup \{\theta_0\}$.

Proof. This random process $(x_1(t;\omega) - \hat{x}_1(t;\theta;\omega))$ is a zero-mean Gaussian process; furthermore, existence of $P_1^0(\theta)$ (see 18)) implies that it is also stationary. It follows therefore [25, p. 69] that $(x_1(t;\omega) - \hat{x}_1(t;\theta;\omega))$ is ergodic if

$$\int_0^\infty ||R(\tau)||d\tau < \infty$$

where

$$R(\tau) = E\{[x_1(t+\tau;\omega) - \hat{x}_1(t+\tau;\theta;\omega)]\ [x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^*\}.$$

From (10), (17) and linear system theory, we have

$$\begin{aligned} R(\tau) &= P_1^0(\theta) \quad \text{for} \quad \tau = 0 \\ &= C'(\exp(A'\tau))\, L(\theta;\theta_0)C'^* \quad \text{for} \quad \tau > 0. \end{aligned}$$

For any $\tau > 0$ we have

$$||R(\tau)|| < ||C'||^2 \cdot ||L(\theta;\theta_0)|| \cdot ||\exp(A'\tau)||.$$

Since A' is stable $\forall\theta \in \Theta$,

$$||\exp(A'\tau)|| < \exp(\lambda\tau) \quad (\forall\theta \in \Theta)$$

where $\lambda < 0$ is the largest eigenvalue of $(A' + A'^*)$. Hence it follows that

$$\int_0^\infty ||R(\tau)||d\tau \leq ||C'||^2 \cdot ||L(\theta;\theta_0)|| \int_0^\infty e^{\lambda\tau}d\tau$$
$$< \infty. \square$$

Remark 1. Following a procedure similar to the one in the proof of Lemma 1, it also follows that the processes $x_1(t;\omega)$ and $\hat{x}_1(t;\theta;\omega)$ are ergodic for $t \geq 0$ and for every $\theta \in \Theta \cup \{\theta_0\}$ assuming the hypothesis of Lemma 1.

From (3), (8), (14) and (15), we have

$$q(\theta;y(\cdot;\omega);T) = (1/T)\left[\int_0^T [(GG^*)^{-1}\hat{x}_1(t;\theta;\omega), \hat{x}_1(t;\theta;\omega)]dt - 2\int_0^T [(GG^*)^{-1}\hat{x}_1(t;\theta;\omega), dy(t;\omega)]\right]. \tag{19}$$

Substitute for $dy(t;\omega)$ from (2) in (19) and after some algebraic manipulations we have

$$q(\theta;y(\cdot;\omega);T) = (1/T)\left[\int_0^T \mathrm{tr}\, G_s[x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)][x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^*G_s dt - \int_0^T [(GG^*)^{-1}x_1(t;\omega),x_1(t;\omega)]dt - \int_0^T [(GG^*)^{-1}\hat{x}_1(t;\theta;\omega),GdW_2(t;\omega)]\right] \tag{20}$$

where G_s denotes the positive "square-root" of $(GG^*)^{-1}$ [29, p. 94], i.e., $G_sG_s = (GG^*)^{-1}$ and G_s is symmetric positive definite. From [1, Chapter 8], expectation of (20) exists; therefore, on using Fubini's theorem [26] we have

$$E\{q(\theta;y(\cdot;\omega);T)\} = (1/T)\left[\int_0^T \mathrm{tr}\, G_sP_1^0(t;\theta)G_s dt - \int_0^T E\{[(GG^*)^{-1}x_1(t;\omega),x_1(t;\omega)]\}dt\right] \tag{21}$$

where we have used the fact that [27, p. 437]

$$E\left\{\int_0^T [(GG^*)^{-1}\hat{x}_1(t;\theta;\omega),GdW_2(t;\omega)]\right\} = 0 \quad (\forall T < \infty)$$

since $W_2(t)$ is an independent increment zero-mean process. Now let $T\to\infty$ and note that all processes involved are asymptotically stationary. Then

$$q(\theta) \triangleq \lim_{T\to\infty} E\{q(\theta;y(\cdot;\omega);T)\}$$

$$= \mathrm{tr}\, G_s P_1^0(\theta) G_s + \lim_{t\to\infty} E\{[(GG^*)^{-1} x_1(t;\omega), x_1(t;\omega)]\}$$

$$= \mathrm{tr}\, G_s P_1^0(\theta) G_s + \mathrm{tr}\, G_s [C_{\theta_0} \quad 0] L(\theta;\theta_0) [C_{\theta_0} \quad 0]^* G_s . \tag{22}$$

Lemma 2. Under the hypothesis of Lemma 1,

$$\lim_{T\to\infty} q(\theta;y(\cdot;\omega)T) = q(\theta) \quad \text{a.s.} \quad (\forall \theta \in \Theta \cup \{\theta_0\}).$$

Proof. From the ergodic theorem [25, p. 68] and Lemma 1, we have

$$\lim_{T\to\infty} (1/T) \int_0^T \mathrm{tr}\, G_s[x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)][x_1(t;\omega) - \hat{x}_1(t;\theta;\omega)]^* G_s \, dt$$

$$= \mathrm{tr}\, G_s P_1^0(\theta) G_s \quad \text{a.s.} \tag{23}$$

Similarly from the ergodic theorem and Remark 1, we have

$$\lim_{T\to\infty} (1/T) \int_0^T [(GG^*)^{-1} x_1(t;\omega), x_1(t;\omega)] dt$$

$$= \mathrm{tr}\, G_s [C_{\theta_0} \quad 0] L(\theta;\theta_0) [C_{\theta_0} \quad 0]^* G_s \quad \text{a.s.} \tag{24}$$

Finally, since $||E\{\hat{x}_1(t;\theta;\omega)\hat{x}_1^*(t;\theta;\omega)\}|| < \infty$ $(\forall \theta \in \Theta \cup \{\theta_0\})$, it follows that [24]

$$\lim_{T\to\infty} (1/T) \int_0^T [(GG^*)^{-1} \hat{x}_1(t;\theta;\omega), G dW_2(t;\omega)] = 0 \quad \text{a.s.} \tag{25}$$

Now from (22) - (25) the desired result follows.□

Remark 2. In the sequel we shall use the stationary version of the likelihood functional for which Lemma 2 holds. It should be noted however that a.s. convergence of the likelihood functional to its stationary version as $T\to\infty$ can be shown to hold following the procedure given in [16, Lemma A] for the discrete-time version. We shall ignore this technical point in this note.

Theorem 1 defines the "limit set" of the ML estimate as $T\to\infty$.

Theorem 1. Suppose that conditions (C1) - (C2) are satisfied. Then the ML estimate of θ on Θ belongs to a set Θ_ℓ w.p.1. as $T\to\infty$ where

$$\Theta_\ell = \{ \theta : \theta \in \Theta,\ q(\theta) = \min_{\theta\in\Theta} q(\theta)\}.$$

Proof. We have, for any $\theta' \in \Theta_\ell$,

$$\lim_{T\to\infty} [H(\theta;y(\cdot;\omega);T)/H(\theta';y(\cdot;\omega);T)]$$

$$= \exp\left[- \lim_{T\to\infty} \frac{T}{2} \left(q(\theta;y(\cdot;\omega);T) - q(\theta';y(\cdot;\omega);T)\right)\right] . \tag{26}$$

From Lemma 2, given any arbitrarily small positive number δ_1, for every ω in a set Λ of probability measure one and for every θ in the finite set Θ, we have

$$|q(\theta;y(\cdot;\omega);T) - q(\theta)| < \delta_1 \quad \text{for all} \quad T > T(\delta_1) \ . \tag{27}$$

Therefore, it follows that $\forall\theta \in \Theta$,

$$-2\delta_1 + q(\theta) - q(\theta') < q(\theta;y(\cdot;\omega);T) - q(\theta';y(\cdot;\omega);T) < 2\delta_1 + q(\theta) - q(\theta')$$
$$(\forall T > T(\delta_1)) \quad (\forall\omega \in \Lambda) \ . \tag{28}$$

Now let

$$\delta_2 = \min_{\theta \notin \Theta_\ell,\ \theta\in\Theta} [q(\theta) - q(\theta')] \ . \tag{29}$$

Clearly $\delta_2 > 0$; choose $\delta_1 < \delta_2/2$. Then from (28) and (29), we have

$$q(\theta;y(\cdot;\omega);T) - q(\theta';y(\cdot;\omega);T) > \delta_2 - 2\delta_1 > 0 \tag{30}$$

for all $T > T(\delta_1)$, $(\forall\omega \in \Lambda)$, and $(\forall\theta \notin \Theta_\ell\ \ \theta \in \Theta)$. Hence from (26) and (30), it follows that, for $\theta' \in \Theta_\ell$,

$$\lim_{T\to\infty} \max_{\theta\notin\Theta_\ell\ \theta\in\Theta} [H(\theta;y(\cdot;\omega);T)/H(\theta';y(\cdot;\omega);T)] = 0 \quad \text{w.p.1.} \tag{31}$$

However,

$$\lim_{T\to\infty} \max_{\theta\in\Theta} [H(\theta;y(\cdot;\omega);T)/H(\theta';y(\cdot;\omega);T)] = 1 \quad \text{w.p.1.} \tag{32}$$

From (31) and (32), the ML estimate of θ on Θ belongs to Θ_ℓ w.p.1. as $T \to \infty$. □

Remark 3: It should be noted that Theorem 1 does not imply convergence of the ML estimate of θ to some θ' in Θ_ℓ. It is possible that the ML estimate might "oscillate" among various θ' in Θ_ℓ.

Now we turn to the question whether the set Θ_ℓ contains the true parameter vector θ_0 when $\theta_0 \in \Theta$. Lemma 3 shows then $q(\theta) \geq q(\theta_0)$; therefore, $\theta_0 \in \Theta_\ell$ whenever $\theta_0 \in \Theta$.

Lemma 3: Suppose that conditions (C1) - (C2) are satisfied. Then $q(\theta) \geq q(\theta_0)$ $\forall\theta \in \Theta$.

Proof: Let $p^T_{W_2}$ be the probability measure induced by the Wiener process $W_2(t;\omega)$ on the space of continuous functions $C = C^m[0,T]$. Furthermore, let $p^T_{y_1,\theta}$ be the probability measure induced by $y_1(t;\omega) \triangleq (GG^*)^{-1} y(t;\omega)$ on $\mathcal{C}$ assuming that $y(t;\omega)$, $0 \leq t \leq T$, results from model (1) - (2) with θ as the "true" parameter. As shown in [1, Chapter 8], $p^T_{y_1,\theta}$ is absolutely continuous w.r.t. $p^T_{W_2}$. Now since both $y_1(t;\omega)$ and $W_2(t;\omega)$ are Gaussian processes, it follows from the Feldman-Hajek dichotomy theorem [25, Chapter 6] that $p^T_{y_1,\theta}$ and $p^T_{W_2}$ are mutually absolutely continuous for any θ in $\Theta \cup \{\theta_0\}$; consequently, $p^T_{y_1,\theta}$ and derivative of $p^T_{y_1,\theta}$ w.r.t. $p^T_{y_1,\theta_0}$ is given by

$$dp^T_{y_1,\theta}/dp^T_{y_1,\theta_0} = H(\theta;y(\cdot;\omega);T)/H(\theta_0;y(\cdot;\omega);T) \triangleq \exp(v_T). \tag{33}$$

From Jensen's inequality [26], we have

$$E\{v_T\} \leq \ln E\{\exp(v_T)\} \quad (\forall T < \infty) \ . \tag{34}$$

Now we have

$$E\{\exp(v_T)\} = \int \left(dp^T_{y_1,\theta} / dp^T_{y_1,\theta_0} \right) dp^T_{y_1,\theta_0} = 1 \quad (\forall T < \infty) \tag{35}$$

which, together with (34), implies that

$$E\{v_T\} \leq 0 \quad (\forall T < \infty) \ . \tag{36}$$

Now from (8) and (22), it follows that

$$\lim_{T\to\infty} E\left\{ -\frac{2}{T} v_T \right\} = q(\theta) - q(\theta_0) \tag{37}$$

which, along with (36), implies that $q(\theta) - q(\theta_0) \geq 0$.□

In light of Lemma 3, Theorem 1 is now modified as follows, for the case $\theta_0 \in \Theta$.

Theorem 2: Suppose that conditions (C1) - (C2) are satisfied and $\theta_0 \in \Theta$. Then the ML estimate of θ on Θ belongs to a set Θ^0_ℓ w.p.1. as $T \to \infty$, where

$$\Theta^0_\ell = \{\theta : \theta \in \Theta \text{ and } q(\theta) = q(\theta_0)\}.$$

Remark 5: Any sufficient condition ensuring consistency of the estimate may be called an "Identifiability Condition." From Theorem 2, a sufficient condition is provided by the inequality

$$q(\theta) \neq q(\theta_0) \quad (\forall \theta \in \Theta) \quad (\theta \neq \theta_0) \ . \tag{38}$$

We also note that our Identifiability Condition is weaker than Balakrishnan's [1] (for the case when $u(t) \equiv 0$ in [1]) in that it is implied by the condition of [1]. From (21), (22) and (38), the Identifiability Condition (38) can also be written as

$$\operatorname{tr} P^0_1(\theta) \neq \operatorname{tr} P^0_1(\theta_0) \quad (\forall \theta \in \Theta) \quad (\theta \neq \theta_0) \ . \tag{39}$$

Model Approximation via ML Identification

Now we focus attention on the case when $\theta_0 \notin \Theta$. Theorem 1 defines the limit set of the ML estimate of θ on Θ as $T \to \infty$. The limit set Θ_ℓ may be redefined as follows

$$\Theta_\ell = \{\theta : \theta \in \Theta,\ q(\theta) = \min_{\theta \in \Theta} q(\theta) - q(\theta_0)\} \ . \tag{40}$$

From the proof of Lemma 3, we have

$$q(\theta) - q(\theta_0) = \lim_{T\to\infty} (2/T)\ E\{\ln[H(\theta_0;y(\cdot;\omega);T)/H(\theta;y(\cdot;\omega);T)]\}$$

$$= \lim_{T\to\infty} (2/T)\ E\{\ln\,[dp^T_{y_1,\theta_0} / dp^T_{y_1,\theta}]\}$$

$$= \lim_{T\to\infty} (2/T)\, E\{\ln [dp^T_{y,\theta_0}/dp^T_{y,\theta}] \qquad (41)$$

The quantity $E\{\ln [dp^T_{y,\theta_0}/dp^T_{y,\theta}]\}$ may be viewed as the continuous-time analog of Kullback information measure for discriminating in favor of the hypothesis, $\theta = \theta_0$, against another hypothesis, $\theta = \theta$, given $y(\zeta;\omega)$, $0 \le \zeta \le T$, under $\theta = \theta_0$. Let

$$J_T(\theta_0;\theta) = E\{\ln [dp^T_{y,\theta_0}/dp^T_{y,\theta}]\} \qquad (42)$$

Then, from (41) and (42), we have

$$\lim_{T\to\infty} \frac{1}{T} J_T(\theta_0;\theta) = \frac{1}{2} (q(\theta) - q(\theta_0)) = \frac{1}{2} \operatorname{tr} G_s[P^0_1(\theta) - P^0_1(\theta_0)]G_s. \qquad (43)$$

Therefore, from (40) and (43), the ML identification techniques leads to the selection of a model $\theta \in \Theta$ which minimizes the asymptotic per unit time (continuous-time analog of) Kullback information measure given by (42)-(43). The Kullback and related information measures have been used in the analysis of parameter estimation and model identification techniques for discrete systems [18-20].

III. CONCLUDING REMARKS

Some results in system identification and model approximation on finite parameter sets pertaining to discrete-time systems have been extended to continuous-time linear dynamic stochastic (Gauss-Markov) systems. All the results obtained were "expected", but analysis involved is technically harder because a white noise process is not so nicely behaved as a white noise sequence.

Specifically, we proved a.s. convergence of the ML estimates of the unknown parameters of a continuous-time linear system. The situation where the true parameter does not belong to the unknown parameter set was also considered. An asymptotic per unit time continuous-time analog of Kullback information measure was proposed for analyzing model approximation via ML identification.

We assumed that the measurement noise covariance was known. The case when it is unknown creates technical problems not present in the discrete-time version [3]. The problem can be (partially) resolved by considering a modified likelihood functional as in [3].

REFERENCES

[1] A. V. Balakrishnan, Stochastic Differential Systems I, Lecture Notes in Economics and Mathematical Systems, Berlin: Springer-Verlag, 1973.
[2] A. V. Balakrishnan, J. Opt. Theory Appl., Vol. 9, pp. 187-212, 1972.
[3] A. Bagchi, Automatica, Vol. 11, pp. 533-536, Sept. 1975.
[4] A. Bagchi, Ph.D. Dissertation in Engineering, University of California, Los Angeles, 1975.
[5] A. Bagchi, "Consistent estimates of parameters in continuous time systems," presented at the IMA Conf. on Analysis and Optimization of Stochastic Systems, Oxford, England, Sept. 1978.
[6] K. W. Iliff and R. E. Maine, J. Guidance & Control, Vol. 2, pp. 228-234, May-June, 1979.

[7] K. J. Åström and P. Eykoff, Automatica, Vol. 7, pp. 123-162, 1971.

[8] R. K. Mehra and D. G. Lainiotis (eds.), System Identification: Advances and Case Studies, New York: Academic Press, 1976.

[9] G. C. Goodwin and R. L. Payne, Dynamic System Identification, New York: Academic Press, 1977.

[10] Special Issue on System Identification and Time-Series Analysis, IEEE Trans. Automatic Control, Vol. AC-19, Dec. 1974.

[11] D. G. Lainiotis, IEEE Trans. Automatic Control, Vol. AC-16, pp. 160-170, April 1971.

[12] D. G. Lainiotis, International Journal Control, Vol. 14, pp. 1137-1148, 1971.

[13] D. G. Lainiotis, Proc. IEEE, Vol. 64, pp. 1126-1143, Aug. 1976.

[14] H. Crámer, Mathematical Methods of Statistics, Princeton, N.J.: Princeton Univ. Press, 1946.

[15] A. Wald, Annals of Math. Statistics, Vol. 20, pp. 595-601, 1949.

[16] J. Rissanen and P.E. Caines, Annals of Statistics, Vol. 7, pp. 297-315, March 1979.

[17] E. Tse and J. J. Anton, IEEE Trans. Automatic Control, Vol. AC-17, pp. 637-646, Oct. 1972.

[18] Y. Baram and N. R. Sandell, Jr., IEEE Trans. Automatic Control, Vol. AC-23, pp. 61-66, Feb. 1978.

[19] R. M. Hawkes and J. B. Moore, Proc. IEEE, Vol. 64, pp. 1143-1150, Aug. 1976.

[20] B. D. O. Anderson, J. B. Moore, and R. M. Hawkes, Automatica, Vol. 14, pp. 615-622, Nov. 1978.

[21] L. Ljung, "On consistency of prediction error identification methods," in System Identification: Advances and Case Studies, R. K. Mehra and D. G. Lainiotis, (eds.), New York: Academic, 1976.

[22] P. E. Caines, IEEE Trans. Automatic Control, Vol. AC-21, pp. 500-505, Aug. 1976.

[23] Y. Baram and N. R. Sandell, Jr., IEEE Trans. Automatic Control, Vol. AC-23, pp. 451-454, June 1978.

[24] R. Z. Khazminskii, "Ergodic properties of recurrent diffusion processes and stabilization of the solution to the Cauchy problem for parabolic equations," Theory Prob. Appl., Vol. 5, 1960.

[25] E. Wong, Stochastic Processes in Information and Dynamical Systems, New York: McGraw-Hill, 1971.

[26] R. B. Ash, Real Analysis and Probability, New York: Academic Press, 1972.

[27] J. L. Doob, Stochastic Processes, New York: Wiley, 1953.

[28] S. Kullback, Information Theory and Statistics, New York: Wiley, 1959.

[29] R. Bellman, Introduction to Matrix Analysis, New York: McGraw-Hill, 1960.

PARTIAL REALIZATIONS OF LINEAR STOCHASTIC TIME-VARYING SYSTEMS

ROBERT E. SKELTON
Purdue University
West Lafayette, IN

ABSTRACT

Definitions are given for minimal-realizations with respect to quadratic model error functions. The theory treats constant or time-varying stochastic linear systems with finite or infinite-time model error functions. A procedure is given which is useful for either model reduction or reduced controller design for time-varying linear systems. In either case, the order of the reduced system is found which yields a prespecified model error index. An algorithm is given to automate the procedure.

1.0 INTRODUCTION

There is much literature available for model reduction of linear time-invariant systems. One can choose among many frequency domain or time domain methods for this purpose. However, quite often problems arise in which time-invariant methods are inappropriate. Examples include circumstances in which;

a) *the model must be most accurate at a specific time or time interval, during the system operation.* Such circumstances include systems which go through "critical" times, and perhaps less model accuracy is required at other times. Examples of critical times t_c which require most model fidelity are:

- spacecraft reentry time t_c
- nuclear reactor startup or other critical times t_c
- time of acquisition, t_c, of a power substation into a larger power network
- time of docking t_c of two spacecraft
- touchdown time t_c of an aircraft, or the passing of an aircraft or rocket through maximum dynamic pressure.

b) *the system is time-varying* (no examples are needed for this common occurance)

c) *the time interval over which the model needs accuracy is finite.* Examples here include finite time control problems:

- air to air missile intercept; guidance and control of launch vehicles
- rapid repositioning of a space vehicle
- many chemical and industrial processes in transient

Of course, these circumstances (a), (b), (c) are not necessarily mutually exclusive. Also, established model reduction methods such as [1]-[6] might be sufficient for some of the above problems, depending upon the accuracy or performance requirements demanded from the reduced model (or reduced controller). Alternately, one might search for methods which exploit the peculiar circumstances at hand: (a), (b) or (c). Such is the goal of this paper.

Fundamental to the objectives of this paper is the establishment of a "value" or "cost" associated with each component of the system. Generally this requires two steps. The first step is to establish the output vector y composed of those variables of interest in the modeling problem, or those variables we wish to regulate in a control problem. Thus, it is presumed that the models exist for the purpose of accurately modeling $y(t)$ over some interval $0<t\leq T$. Any reduced model generating the approximate output $\hat{y}(t)$ is therefore judged by the value of the model error index $(\mathcal{Q})$,

$$\mathcal{Q} \triangleq \left|\frac{\mathcal{V}-\mathcal{V}_R}{\mathcal{V}}\right| \tag{1.1}$$

where $\mathcal{V}_R$ is the performance value of the reduced model

$$\mathcal{V}_R \triangleq \frac{1}{T}E\left\{\int_0^T \|\hat{y}(t)\|^2_{Q(t)}dt + \|\hat{y}(T)\|^2_{Q(T)}\right\} \tag{1.2}$$

and $\mathcal{Q}$ is normalized by the value of the full system

$$\mathcal{V} \triangleq \frac{1}{T}E\left\{\int_0^T \|y(t)\|^2_{Q(t)}\,dt + \|y(T)\|^2_{Q(T)}\right\} \tag{1.3}$$

where $\|y(t)\|^2_{Q(t)} \triangleq y^T(t)Q(t)y(t)$ and E denotes expected value. Of course the $\mathcal{Q}$ can only be evaluated *after* model reduction. The information available *a priori* is $\mathcal{V}$ and the high order evaluation model generating y. Hence, section 2.0 shows how $\mathcal{V}$ can be decomposed into the sum of *in situ* component costs (associated with each component of the system when all components are in place and interacting). Model reduction decisions can then be based upon this cost decomposition, where components with smaller *in situ* component costs are deleted from this model. Time-invariant versions of a similar theory were described in [6]. However, the present paper offers a *different* definition of component cost than [6]. Ref. [6] considered an *input-induced* cost contribution as opposed to the current *output-induced* cost contribution described in Section 2.

Section 3.0 defines *minimal realizations with respect to quadratic criteria* (1.3) as *the lowest order model with the property* $\mathcal{Q} = 0$. Necessary and sufficient conditions for such realizations are given. One possible advantage of this cost-induced realization theory is that it readily suggests *partial* realizations. Such partial realizations are suggested by the *in situ* component costs which are *scalars* and are therefore easily ranked in order of their magnitudes. This is considerably easier than the *matrix* rank calculations required in computing Kalman's canonical form (required for minimal realizations) or the calculations and additional decisions required in the matching of such moments as residues, Markov parameters, Hankel matrices, etc. Furthermore, the present methods are not restricted to time-invariant systems.

Section 4.0 applies the theory to the reduction of centralized LQG controllers. The resulting controller component cost algorithm (CCCA) iteratively produces dynamic controllers of lower order until a prescirbed controller error index is satisfied. These data allow a systematic tradeoff between controller order and controlled system performance for time varying systems.

2.0 COMPONENT COSTS IN LINEAR SYSTEMS

The "Evaluation Model"

$$\dot{x} = A(t)x + D(t)\,w \qquad x \in R^{n_x}, \quad w \in R^{n_w}$$
$$y = C(t)x \qquad\qquad y \in R^{n_y} \qquad (2.1)$$

is presumed to be an accurate representation of $y(t)$, $0 < t \leq T$, but n_x is presumed too large for analysis. The zero-mean white noise $w(t)$ is uncorrelated with the initial vector $x(o)$ and has intensity $W(t)$. $x(o)$ has zero-mean and covariance $X(o)$. The n_y variables in y are those which must be accurately described over the interval $0 < t \leq T$. Thus, we are interested in truncating components from the evaluation model to form reduced models which make only small perturbations in the function (1.3). In order to determine which components of (2.1) should be deleted from the model, the components must first be described;

$$\dot{x}_i = \sum_{j=1}^{n} A_{ij}(t)x_j + D_i(t)w \quad , \quad x_i \in R^{n_i} \qquad (2.2)$$

$$y = \sum_{j-1}^{n} C_j(t)x_j \qquad i,j = 1,2,\ldots n \qquad (2.3)$$

where x_i is the state of the ith component and $n_x = \sum_{i=1}^{n} n_i$. x_i might represent the state of a *physical* component of the system such as a substation in a power network, or a substructure in a large flexible space structure, etc. Alternately, the components might be chosen for *mathematical* convenience. For example, *after* a transformation of the physical coordinates the components might be defined such that some of the A_{ij}, or D_i or C_i are zero. In any event, the dynamic coupling between the system components i and j is described by the terms $A_{ij}x_j$, $i \neq j$.

It is our task in this section to assign a value to each component (2.2) of the system (2.1), using only *a priori* information. For this purpose we need the following definition:

Definition: *The in situ component cost* V_i *associated with component* i *having state* x_i *is the net effect of* x_i *in the overall system performance value* V*, and* V_i *has these properties:*

$$V_i \triangleq \tfrac{1}{2} E \frac{\partial Y}{\partial x_i} x_i \quad , \quad Y \triangleq \frac{1}{T}\left\{\int_0^T \|y(t)\|^2_{Q(t)}dt + \|y(T)\|^2_{Q(T)}\right\} \qquad (2.4a)$$

$$V = \sum_{i=1}^{n} V_i \qquad (2.4b)$$

From this definition it follows that

$$V_i = \frac{1}{T}\left\{\mathrm{tr}\int_0^T [X(t)C^T(t)Q(t)C(t)]_{ii}dt + \mathrm{tr}[X(T)C^T(T)Q(T)C(T)]_{ii}\right\} \qquad (2.4c)$$

where X is the state covariance matrix satisfying

$$\dot{X} = XA^T + AX + DWD^T \quad , \quad X(0) = X_0$$

and $[\cdot]_{ii}$ is the $n_i x n_i$ matrix partition of $[\cdot]$. (2.4d)

We now arrange the components (2.2) in order of their component costs V_i so that the arrangement

$$x^T = (x_1^T, x_2^T, \ldots x_r^T, x_{r+1}^T, \ldots x_n) \triangleq (x_R^T, x_T^T) \qquad (2.5)$$

implies

$$V_1 \geq V_2 \geq \ldots \geq V_r > V_{r+1} \geq \ldots \geq V_n \qquad (2.6)$$

If r denotes the number of components which are to be retained in the model and x_R and x_T are defined by (2.5) then the system (2.1) can be described by

$$\begin{pmatrix}\dot{x}_R\\ \dot{x}_T\end{pmatrix} = \begin{bmatrix}A_R & A_{RT}\\ A_{TR} & A_T\end{bmatrix}\begin{pmatrix}x_R\\ x_T\end{pmatrix} + \begin{bmatrix}D_R\\ D_T\end{bmatrix} w \ , \quad y = [C_R \ \ C_T]\begin{pmatrix}x_R\\ x_T\end{pmatrix} \qquad (2.7a)$$

where in these *rearranged* coordinates (2.5),

$$A_R = \begin{bmatrix}A_{11} & A_{12} & \\ A_{21} & A_{22} & \\ & & \ddots \\ & & & A_{rr}\end{bmatrix} \quad A_{RT} = \begin{bmatrix}A_{1,r+1} & A_{1,r+2} & \\ A_{2,r+1} & A_{2,r+2} & \\ & & \ddots & A_{rn}\end{bmatrix}, \quad D_R = \begin{bmatrix}D_1\\ \vdots\\ D_n\end{bmatrix},$$

$$A_{TR} = \begin{bmatrix}A_{r+1,1} & A_{r+1,2} & \\ A_{r+2,1} & A_{r+2,2} & \\ & & \ddots & A_{nr}\end{bmatrix}, \quad A_T = \begin{bmatrix}A_{r+1,r+1} & A_{r+1,r+2} & \\ A_{r+2,r+1} & A_{r+2,r+2} & \\ & & \ddots & A_{nn}\end{bmatrix},$$

$$D_R = \begin{bmatrix}D_{r+1}\\ \vdots\\ D_n\end{bmatrix}, \quad X = \begin{bmatrix}\hat{X}_R & \hat{X}_{RT}\\ \hat{X}_{RT}^T & \hat{X}_T\end{bmatrix}, \quad C_R \triangleq [C_1 C_2 \ldots C_R] , \qquad (2.7B)$$

$$C_T \triangleq [C_{r+1} C_{r+2} \ldots C_n]$$

The partial realization is

$$\dot{\hat{x}}_R = A_R\hat{x}_R + D_R w \ , \quad y = C_R\hat{x}_R \qquad (2.8)$$

where $\hat{x}_R$ is the state of the partial realization and must be distinguished (thus, the use of the ^ in (2.8)) from the corresponding sub-state x_R of the complete model (2.7). We assign to (2.8) the initial values $\hat{x}_R(0) = x_R(0)$. Hence, $E[\hat{x}_R(0)\hat{x}_R^T(0)] = \hat{X}_R(0)$. $\hat{y}$ is the output of the partial realization (2.8.).

Now suppose we again apply the component cost notions (2.4) to the system (2.7) except now let the components be only two: x_R and x_T. This is the *macroscopic* view of the cost-ordered system, whereas (2.2) presents a *microscopic* view of the system. Applying (2.4) for n = 2 to (2.7) yields, usng slightly different notation,

$$V = \hat{V}_R + \hat{V}_T \ , \quad \hat{V}_R = \sum_{i=1}^{r} V_i \ , \quad \hat{V}_T = \sum_{i=r+1}^{n} V_i \qquad (2.9)$$

where $\hat{V}_R$ and $\hat{V}_T$ satisfy, using the notation in (2.7),

$$\hat{V}_R = \frac{1}{T}\mathrm{tr}\left\{\int_0^T [\hat{X}_R(t)C_R^T(t)Q(t)C_R(t) + \hat{X}_{RT}(t)C_T^T(t)Q(t)C_R(t)]dt\right\}$$

$$+ [\hat{X}_R(T)C_R^T(T)Q(T)C_R(T) + \hat{X}_{RT}(T)C_T^T Q(T)C_R(T)] = \sum_{i=1}^{r} V_i \qquad (2.10)$$

$$\hat{\mathcal{V}}_T = \frac{1}{T}\mathrm{tr}\left\{\int_0^T [\hat{X}_{RT}^T(t)C_R^T(t)Q(t)C_T(t) + \hat{X}_T(t)C_T^T(t)Q(t)C_T(t)]dt + [\hat{X}_{RT}^T(T)C_R^T(T)Q(T)C_T(T) + \hat{X}_T(T)C_T^T(T)Q(T)C_T(T)] = \sum_{i=r+1}^{n} \mathcal{V}_i \right. \tag{2.11}$$

which follows directly by setting the subscripts 1=R and 2=T in (2.4). Thus the fractional part of the cost associated with the truncated components provides an *a priori* prediction of the model error associated with (2.8). We label this fractional part of the cost $\hat{\mathcal{Q}}$ and call it the "predicted model error index",

$$\hat{\mathcal{Q}} = \left|\frac{\hat{\mathcal{V}}_T}{\mathcal{V}}\right| = \left|\frac{\mathcal{V}-\hat{\mathcal{V}}_R}{\mathcal{V}}\right| , \quad 0 \le \hat{\mathcal{Q}} \le 1 \tag{2.12}$$

whereas, the *actual* model error index $\mathcal{Q}$ as given by (1.1) must be computed *after* model reduction. Note that $\mathcal{Q}$ lies in the range $0 \le \mathcal{Q} \le \infty$, whereas the range of $\mathcal{Q}$ is telescoped to $0 \le \hat{\mathcal{Q}} \le 1$. The better the model (2.8) the closer is $\hat{\mathcal{Q}}$ to zero. If $\mathcal{V} > \mathcal{V}_R$ and $\mathcal{V} \geq \hat{\mathcal{V}}_R$, the error in the prediction $\hat{\mathcal{Q}}$ is

$$\tilde{\mathcal{Q}} \triangleq \mathcal{Q} - \hat{\mathcal{Q}} = \frac{1}{\mathcal{V}}(\hat{\mathcal{V}}_R - \mathcal{V}_R) \tag{2.13}$$

To use component cost analysis to construct a reduced model which is accurate within a specified model error index $\bar{\mathcal{Q}}$, one would preceed as follows.

The CCA Algorithm for Model Reduction of Time-Varying Systems:

STEP 0: Input data: (A,D,C,Q,W) and either r or $\bar{\mathcal{Q}}$.

STEP I: Using (2.4c), (2.4d), compute and order the component costs $\mathcal{V}_i$. If r is specified skip to Step III.

STEP II: If r is not specified, find the r which satisfies

$$\hat{\mathcal{Q}} \triangleq \sum_{i=r+1}^{n} \mathcal{V}_i/\mathcal{V} \le \mathcal{Q} \quad (\bar{\mathcal{Q}} \text{ specified} << 1) \tag{2.14}$$

STEP III: Reorder the system components x_i according to the order of their component costs $\mathcal{V}_i$. This gives (2.7).

STEP IV: Compute

$$\mathcal{V}_R = \frac{1}{T}\mathrm{tr}\left\{\int_0^T X_R(t)C_R^T(t)Q(t)C_R(t)dt + X_R(T)C_R^T(T)Q(T)C_R(T)\right. \tag{2.15}$$

$$\dot{X}_R = X_R A_R^T + A_R X_R + D_R W D_R^T , \quad X_R(0) = \hat{X}_R(0) \tag{2.16}$$

If

$$\mathcal{Q} = \left|\frac{\mathcal{V}-\mathcal{V}_R}{\mathcal{V}}\right| > \bar{\mathcal{Q}} \tag{2.17}$$

set r = r+1 and return to STEP III. Otherwise, stop and print the reduced model $(A_R,D_R,C_R,X_R(0))$ and its model error index $\mathcal{Q}$.

3.0 MINIMAL AND PARTIAL REALIZATIONS WITH RESPECT TO QUADRATIC COSTS

The model reduction and system decomposition techniques of the previous section require no new mathematical tools beyond the standard LQG theory. This section seeks to examine various mathematical properties of the reduced models produced by the CCA algorithm. More specifically we are interested in the questions:

1. Under what condition is the predicted model error index correct ($\hat{Q}=Q$)?
2. Under what condition is the model error index zero ($Q=0$)?
3. Under what condition is the model error index minimized by the CCA algorithm?

To assemble the machinery necessary to answer these questions we need the following definitions.

Definition 1: *The partial realizaiton* $(A_R,D_R,C_R,X_R(0))$ *is said to be a cost-equivalent realization if* $Q=0$.

Definition 2: *The rth order partial realization is said to be a cost-minimal realization if* r *is the smallest integer for which* $Q=0$.

Now suppose $\tilde{X}_R$ is defined by

$$\dot{\tilde{X}}_R = A_R\tilde{X}_R + \tilde{X}_RA_R^T + A_{RT}\hat{X}_{RT}^T + \hat{X}_{RT}A_{RT}^T \quad , \quad \tilde{X}_R(0) = 0 \tag{3.1a}$$

where $\hat{X}_{RT}$ satisfies (2.7b), (2.4d), repeated here in expanded form,

$$\dot{\hat{X}}_{RT} = A_R\hat{X}_{RT} + \hat{X}_{RT}A_T^T + A_{RT}\hat{X}_T + \hat{X}_RA_{TR}^T + D_RWD_T^T \tag{3.1b}$$

$$\dot{\hat{X}}_T = A_T\hat{X}_T + \hat{X}_TA_T^T + A_{TR}\hat{X}_{RT} + \hat{X}_{RT}^TA_{TR}^T + D_TWD_T^T \tag{3.1c}$$

where $\hat{X}_{RT}(0)$ and $\hat{X}_T(0)$ are specified in (2.4d).

Question 1 is answered by theorem 1.

Theorem 1: *Let* $(A_R,D_R,C_R,\hat{X}_R(0))$ *be a partial realization of* (2.1). *The predicted model error index* $\hat{Q}$ *is exact in the sense that* $\hat{Q}=Q$ *under either of the following conditions* (a),(b),(c), *or* (d);

(a) *if and only if*

$$\operatorname{tr}\left\{\int_0^T[\tilde{X}_R(t)C_R^T(t)Q(t)C_R(t) + \hat{X}_{RT}(t)C_T^T(t)Q(t)C_R(t)]dt \right.$$
$$\left. + \tilde{X}_R(T)C_R^T(T)Q(T)C_R(T) + \hat{X}_{RT}(T)C_T^T(T)Q(T)C_R(T)\right\} = 0 \tag{3.2}$$

(b) *if* x_T *is unobservable on the interval* $t\varepsilon[0,T]$,

(c) *if* x_T *is undisturbale on the interval* $t\varepsilon[0,T]$ *and* $\hat{X}_T(0)=0,\hat{X}_{RT}(0)=0$,

(d) *if* $\hat{X}_{RT}(t) \equiv 0$.

Proof: The proof assumes $V>V_R$, $V>\hat{V}_R$. Otherwise slight modifications are required. Hence, from (2.13) we must show that $V_R=V_R$. Subtract (2.15) from (2.10) to obtain (3.2) directly. Now to prove (b) we may assume without loss of generality that $A_{RT}(t)=0$, $C_T(t)=0$ for $t\varepsilon[0,T]$ since x_T is unobservable. This means that (3.2) is satisfied if $\tilde{X}_R(t)\equiv 0$ and (3.1a) reveals that $\tilde{X}_R(t)\equiv 0$ if $A_{RT}=0$. To prove (c) note that since x_T is undisturbable we may assume $A_{TR}(t)=0$, $D_T(t)=0$, $t\varepsilon[0,T]$. This substitution into (3.1) leads to $X_T(t)=0$ if $\hat{X}_T(0)=0$, and $\hat{X}_{RT}(t)=0$ if $\hat{X}_{RT}(0)=0$. Finally $\hat{X}_{RT}(t)=0$ since

$\hat{X}_{RT}(t)=0$, resulting in the satisfaction of (3.2). To prove (d), set $\hat{X}_{RT}(t)\equiv 0$ in (3.1a) to get $X_R(t)\equiv 0$. Hence, (3.2) is satisfied. #

Question 2 is answered by theorem 2.

Theorem 2: *The partial realization* $(A_R, D_R, C_R, X_R(0))$ *is a cost-equivalent realization of* (2.1);

(a) *if and only if*

$$\frac{1}{T}\mathrm{tr}\left\{\int_0^T [\tilde{X}_R(t)C_R^T(t)Q(t)C_R(t) + 2\hat{X}_{RT}(t)C_T^T(t)Q(T)C_R(t) + \hat{X}_T(t)C_T^T(t)Q(t)C_T(t)]dt + \tilde{X}_R(T)C_R^T(T)Q(T)C_R(T) + 2\hat{X}_{RT}(T)C_T^T(T)Q(T)C_R(T) + \hat{X}_T(T)C_T^T(T)Q(T)C_T(T)\right\} = 0 \tag{3.3}$$

(b) *if* x_T *is unobservable on the interval* $t\varepsilon[0,T)$.

(c) *if* x_T *is undisturbable on the interval* $t\varepsilon[0,T]$ *and* $\hat{X}_T(0)=0, \hat{X}_{RT}(0)=0$.

Proof: To prove (a) we must show that (1.1) yields $\mathcal{Q}=0$ if and only if (3.3) holds. Hence, we must shown that $V=V_R$. Eqs. (2.4c), (2.4d) yield V and (2.15), (2.17) yield V_R. Subtract V_R from V to get the left half side of (3.3), after substituting the definition $\tilde{X}_R=\hat{X}_R-X_R$. To prove (b) and (c) we need only to cite the proof of theorem 1 which showed that $\tilde{X}_R(t)\equiv 0$ and $C_T(t)\equiv 0$ (hence, (3.3) is satisfied) if X_T is unobservable, and that $\tilde{X}_R(t)=0$, $\hat{X}_{RT}(t)=0$, $\hat{X}_T(t)=0$ (hence, (3.3) again is satisfied) if x_T is undistrubable. #

Theorems 1 and 2 suggest (from conditions (b) and (c)) that a realization is cost-minimal when the model error prediction is correct, $\tilde{\mathcal{Q}}=0$. This is not proved, however, since conditions (b) and (c) are only sufficient. Theorem 1 also shows that $\tilde{X}_R(t)\equiv 0$ if x_T is *either* unobservable or undisturbable. Hence, the norm of $\tilde{X}_R$ may prove useful as a measure of the degree observability-disturbability of x_T.

A special choice of coordinates will be of interest to us in the model reduction of sections 2 and 3. These are the coordinates in which $\hat{X}_{RT}(t)\equiv 0$. Theoretically such a set of coordinates can be found as follows.

1. Compute $X(t)$ from (2.4d).
2. Find an orthonormal matrix T such that

$$T(t)^T X(t) T(t) = \text{block diag. } [\hat{X}_R(t), \hat{X}_T(t)] \tag{3.4}$$

The new model parameters are defined by $T(t)^T A(t) T(t)$, $T(t)^T D(t)$, $C(t)T(t)$ where the initial covariance in the new coordinates is $\left[\begin{array}{c|c} \hat{X}_R(0) & 0 \\ \hline 0 & \hat{X}_T(0) \end{array}\right]$.

In these coordinates, the CCA algorithm simplifies by the deletion of (2.17) and is *not* iterative. This simplification is due to the fact that there is no error in the predicted model error index $\hat{\mathcal{Q}}$, by virtue of theorem 1, part (d). The revised CCA algorithm is as follows.

THE COST-DECOUPLED CCA ALGORITHM

STEP 0: Input Data $\{A(t), D(t), C(t), Q(t), Q(T), W(t), X_0, \bar{\mathcal{Q}}.\}$

STEP 1: Compute X(t):

$$\dot{X} = AX + XA^T + DWD^T \quad , \quad X(0) = X_0 \qquad (3.5)$$

STEP 2: Compute T(t) such that

$$T^T(t)X(t)T(t) \triangleq \text{diag.} \quad , \quad T(t)^T T(t) = I \qquad (3.6)$$

STEP 3: Redefine and store

$$T^T(t)A(t)T(t) - T(t)^T\dot{T}(t) \quad \rightarrow \quad A(t) \qquad (3.7)$$

$$T^T(t)D(t) \quad \rightarrow \quad D(t) \qquad (3.8)$$

$$C(t)T(t) \quad \rightarrow \quad C(t) \qquad (3.9)$$

$$T^T(0)X(0)T(0) \quad \rightarrow \quad X(0) \qquad (3.10)$$

$$T^T(t)X(t)T(t) \quad \rightarrow \quad X(t) \qquad (3.11)$$

STEP 4: Compute Component Costs

$$\mathcal{V}_i = \frac{1}{T}\text{tr}\left[\int_0^T X_{ii}(t)C_i^T(t)Q(t)C_i(t)dt + X_{ii}(T)C_i^T(T)Q(t)C_i(T)\right] \qquad (3.12)$$

where the subscripted matrices X_{ii} denotes the $n_i x n_i$ block diagonal element of X, and C_i denotes the kxn_i matrix associated with the approiate n_i columns of C.

STEP 5: Find r such that

$$\mathcal{Q} = \frac{\sum_{i=r+1}^{n} \mathcal{V}_i}{\sum_{i=1}^{n} \mathcal{V}_i} \quad \leq \quad \overline{\mathcal{Q}} \qquad (3.13)$$

STEP 6: The rth order partial realization is $\{A_R(t), D_R(t), C_R(t), X_R(0)\}$ where $A_R(t)$ is obtained from A(t) by deleting those rows and columns with the n-r smallest $\mathcal{V}_i$, $D_R(t)$ is obtained by deleting the corresponding rows from D(t), and $C_R(t)$ is obtained by deleting the corresponding columns from C(t).

An important property of this algorithm is given by

Theorem 3: *The cost-decoupled CCA algorithm truncates those components* x_i *which cause the actual model error index* $\mathcal{Q}$ *to be minimized.*

Proof: In coordinates defined by (3.6), the component cost (2.4c) reduces to (3.12). Hence $\mathcal{V}_i$ is independent of $\mathcal{V}_j$, $i \neq j$. From Theorem 1, part (d) we know that

$$\hat{\mathcal{Q}} = \mathcal{Q} = \frac{1}{\mathcal{V}}(\mathcal{V} - \sum_{i=1}^{r} \mathcal{V}_i) = \frac{1}{\mathcal{V}}\sum_{i=r+1}^{r} \mathcal{V}_i .$$

Since the CCA algorithm has the property

$$\mathcal{V}_i \geq \mathcal{V}_j \;\; \forall \; i\varepsilon R, \; j\varepsilon T$$

where $i\varepsilon R$ denotes those set of indices associated with the r largest component costs $\mathcal{V}_i$, and $j\varepsilon T$ denotes those set of indices associated with the n-r smallest component costs $\mathcal{V}_j$.

Then it immediately follows that

$$\Sigma V_i \leq \Sigma V_j$$

for $i \varepsilon R$ and for any n-r choices of $j \notin R$ #

4.0 APPLICATION OF CCA TO THE DESIGN OF REDUCED ORDER CONTROLLERS

In this sectin we assume that the assigned task is to design a linear dynamical controller of order $r<n$. A traditional approach is to first use (one's favorite) model reduction scheme and reduce the model to order n, and then to design the controller which is optimal for the reduced model. In this sequence of events the tools of model reduction are used first and the tools of optimal control are used second. In the approach proposed in the sequel, just the opposite sequence of events occur. The tools of optimal control are applied first and the tools of model reduction are applied second. The reasoning for this sequence is that open-loop model reduction may ignore some important coupling of coordinates in the subsequent control feedback loops, [7]. Of course, the computational burdens are greater. Hence, the following procedure is useful only after model has been reduced (by open-loop methods such as in sections 2 or 3 to "Riccati-Solvable" dimension n_r associated with an off-line computer (to be used only in the controller *design* process.) The on-line computer limitations are presumed more severe so that the on-line controller dimension is smaller still, $r<n_r<n$.

The design of the reduced controller follows these steps.

The Controller Component Cost Algorithm (CCCA)

1. Find the optimal controller for

$$x = A(t)x + B(t)u + D(t)w \tag{4.1a}$$

$$y = C(t)x \quad , \quad z = M(t)x+v$$

$$V = \frac{1}{T} E tr \left\{ \int_0^T \left(\|y(t)\|^2_{Q(t)} + \|u(t)\|^2_{R(t)} \right) dt + \|y(T)\|^2_{Q(T)} \right\} \tag{4.1b}$$

where z are the available measurements corrupted by white noise $V \sim N(0,V)$.

2. Write the closed loop optimal system in the form (1.3), (2.1) by proper interpretation of matrices:

$$\left[\begin{array}{c|c} A & BG \\ \hline FM & A+BG-FM \end{array}\right] \rightarrow A \quad , \left[\begin{array}{c|c} D & 0 \\ \hline 0 & G \end{array}\right] \rightarrow D \, , \left[\begin{array}{c|c} C & 0 \\ \hline 0 & G \end{array}\right] \rightarrow C \quad , \begin{pmatrix} y \\ u \end{pmatrix} \rightarrow y \quad , \left[\begin{array}{c|c} Q & 0 \\ \hline 0 & R \end{array}\right] \rightarrow Q$$

where G and F are respectively the control and filter gains associated with problem (4.1)., and the new state is $(x^T, \hat{x}^T)$.

3. Apply the CCA algorithm of section 2. The controller states $\hat{x}_i$ with the smallest component cost V_i will be truncated. The reduced controller is obtained from the truncated $\hat{x}$ equations.

The CCA algorithm of section 2 is used here instead of section 3, since we wish to truncate only $\hat{x}$ components and not a linear combination of x and $\hat{x}$.

5.0 CONCLUSION

The theory of component cost analysis (CCA) has been briefly described for time-varying systems. This analysis determines the *in situ* contribution of each system component in the performance functional. CCA, therefore determines which system components are "critical" in terms of dominating the performance. This information can possibly be used for two different purposes:

- redesign the critical system components

or,

- retain only the critical system component models in model reduction decisions

An algorithm is given to automate the CCA model reduction (or controller reduction) process to achieve a reduced model with a specified model error index. The proper coordinates for model reduction are defined so that the predicted model error index contains no error, and in fact this model error index has been minimized by the CCA algorithm for these "cost-decoupled" coordinates.

REFERENCES

[1] Kokotovic, P.V. and R.A. Yackel, "Singular Parturbation of Linear Regulators; Basic Theorems," IEEE Trans. Auto. Control, Vol. AC-17, Feb. 1972, pp. 29-37.

[2] Edwin, C.Y., J.V. Medanic, and W.R. Perkins, "Generalized Hessenberg Transformations for Reduced-Order Modeling of Large-Scale System," Int. J. Control, 1978, Vol. 27, No. 4, pp. 493-512.

[3] Hutton, M.F. and B. Friedland, "Routh Approximations for Reducing Order of Linear, Time-Invariant Systems," IEEE Trans. Auto. Control, Vol. AC-20, No. 3, June 1975, pp. 329-337.

[4] Wilson, D.A., Model Reduction for Multivariable Systems," Int. J. Control, Vol. 20, No. 1, 1974, pp. 57-64.

[5] Moore, B.C., "Principal Component Analysis in Linear Systems: Controllability, Observability and Model Reduction," to appear, IEEE Trans. Auto. Control.

[6] Skelton, R.E., "Modal Cost Analysis in Model and Controller Reductions," 17th Allerton Conf. on Communication, Control and Computing, October 1979, pp. 421-434.

[7] Skelton, R.E., "Cost Decomposition of Linear Systems," Int. J. Control, accepted for publication April, 1980.

ACKNOWLEDGEMENTS

Support from the National Science Foundation Grant ENG-7711703 is gratefully acknowledged. Professor Skelton also is a consultant for the Naval Research Lab where much of the motivation for the time-varying problem originated.

MODELING AND PERFORMANCE ANALYSIS OF THE PACKET FORMATION PROCESS IN PACKET-SWITCHED TELECOMMUNICATIONS*

JAMES S. MEDITCH
Department of Electrical Engineering
University of Washington
Seattle, Washington 98195

ABSTRACT

Under physically reasonable assumptions, it is shown that the process of converting messages into packets in a packet-switched network can be modeled as an M/G/1 queueing system. The model is then used to specify the performance characteristics and tradeoffs of the process. An example is included to illustrate the results.

1. INTRODUCTION

An important process in the operation of a packet-switched telecommunications network is that of decomposition of messages into packets, the latter being the standard data element for the transmission of information in such networks [1,2]. At each destination, the packets are then reassembled to obtain the original messages. Typically, user-generated messages enter the network at one of its switching nodes where the process of packet formation takes place. The nodes are, in fact, minicomputers, e.g., the IMP's and TIPS in the ARPANET [1,2], and perform a variety of other processes which include routing, error control, flow control, buffering, sequencing, and packet reassembly into messages.

In this paper, we develop a model for the packet formation process. The model is based on physically reasonable assumptions, and provides simple relationships which define the performance characteristics and tradeoffs of the process.

Our presentation is as follows. We develop the model and its associated performance relationships in Section 2. In Section 3, we illustrate our results via an example, and then conclude in Section 4 with a discussion of the model and also indicate there some issues for future research.

2. PACKET FORMATION PROCESS MODEL

The process with which we are dealing is depicted below in Fig. 1 where the block marked "Packet Formation" includes the queueing of arriving messages which are then serviced on a first-come/first-serve

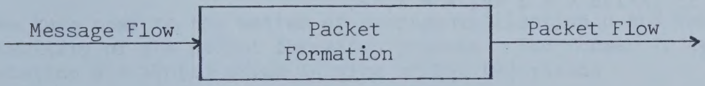


Figure 1. Packet formation process.

---

* Research supported in part by the U.S. Air Force Office of Scientific Research under AFOSR Grant No. 78-3546.

basis to convert them into packets. We assume that the message arrival process is Poisson with an average arrival rate of λ messages/sec, and that the message lengths are independent and exponentially distributed with average length $1/\mu$ bits. We also assume that the arrival times and message lengths are independent of each other.

We consider the decomposition of messages into packets wherein the packet format is as indicated in Fig. 2. The leading and trailing fields of the packet are assumed to be of fixed length M and N bits, respectively, for all packets. These fields will typically be comprised of elements such as clock synchronization bits, control characters for message delimiting, source-destination addresses, protocol information, and hardware and software error checksums. The text field is taken to be variable up to a maximum of B bits. Thus, a message of say 2.7 B bits is assumed to be decomposed into two packets each of length M+N+B bits and one packet of length M+N+0.7 B bits.

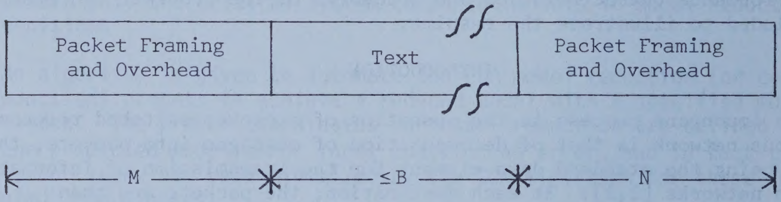


Figure 2. Packet format.

We assume that the time required by the processor to form one packet, including packet framing and overhead, is τ sec, independent of the number of bits in the text field. This is a reasonable assumption since the service time which is defined below will include the effect of segmentation of messages into text fields.

We let $x \geq 0$ denote message length and define y to be the time necessary to convert a message into packets, i.e., y is the service time. From this and the above definition of τ, it follows that

$$y = n\tau \tag{1}$$

where $n = k$ if $(k-1)B < x \leq kB$, $k = 1, 2, \ldots$.

Letting

$$p_k \triangleq P[y = k\tau]$$

where $k = 0, 1, \ldots$ and $P[\cdot]$ denotes the probability, and utilizing the fact that message lengths are exponentially distributed with mean $1/\mu$, it is a direct matter to show that

$$p_k = \begin{cases} 0 & \text{for } k = 0 \\ e^{-(k-1)\mu B}\,(1-e^{-\mu B}) & \text{for } k = 1, 2, \ldots \end{cases} \tag{2}$$

Since the message arrival process is Poisson, it follows that the packet formation process is an M/G/1 queueing system in which the general service process is specified by Eqs. (1) and (2). This permits us to employ a number of well-known queueing theoretic results including the Pollaczek-Khinchin (P-K) mean value formula [1] in order to characterize the properties of this process.

We consider the average service time

$$\overline{y} = \tau \sum_{k=o}^{\infty} kp_k \tag{3}$$

for converting messages into packets. Substituting into Eq. (3) from Eq. (2), we find after some algebra that

$$\overline{y} = \frac{\tau}{1 - e^{-\mu B}} \tag{4}$$

where the units on $\overline{y}$ are sec/message. From Eqs. (1) and (4), it is obvious that the average number of packets per message is

$$\overline{n} = \frac{1}{1 - e^{-\mu B}} \tag{5}$$

In order to utilize the P-K mean value formula, we shall also need the second moment of y which is defined by

$$\overline{y^2} = \tau^2 \sum_{k=o}^{\infty} k^2 p_k$$

Using Eq. (2) again, it is relatively easy to show that

$$\overline{y^2} = \frac{(1 + e^{-\mu B})\,\tau^2}{(1 - e^{-\mu B})^2} \tag{6}$$

We turn next to the matter of average utilization ρ and the issue of the stability of the packet formation process. The former is defined by the relation $\rho = \lambda\overline{y}$ [1] which in view of Eq. (4) yields

$$\rho = \frac{\lambda\tau}{1 - e^{-\mu B}} \tag{7}$$

A necessary and sufficient condition for the packet formation process to be stable in the sense that the queue of messages waiting for service is finite for all time is that $0 \leq \rho < 1$ [1]. As a case in point then, if λ, μ, and B are specified, stability requires that the packet formation time τ satisfy

$$\tau < \frac{1 - e^{-\mu B}}{\lambda} \tag{8}$$

Similar arguments also obviously apply in the case of any one of the above model parameters in terms of the other three.

We are now in a position to apply the P-K mean value formula, Little's result, and the independence of the arrival and service processes to conduct a queueing and average delay analysis of our process.

The average time W that a message spends in queue is given by the P-K formula [1]

$$W = \frac{\lambda \overline{y^2}}{2(1-\rho)}$$

For $\overline{y^2}$ and ρ given by Eqs. (6) and (7), respectively, this becomes

$$W = \frac{\lambda\tau^2(1 + e^{-\mu B})}{2(1 - e^{-\mu B})(1 - e^{-\mu B} - \lambda\tau)} \tag{9}$$

expressed in sec. Little's result then yields the average number of messages in queue as $\overline{N}_q = \lambda W$.

Since the arrival and service processes are independent, the average time that a message spends in queue and service, i.e., its average time in the system, is

$$T = \overline{y} + W$$

From Eqs. (4) and (9), it follows that

$$T = \frac{\tau(2 - \lambda\tau)}{2(1 - e^{-\mu B} - \lambda\tau)} \tag{10}$$

Using Little's result again, we have that the average number of messages in the system is $\overline{N} = \lambda T$.

Beyond its immediate use as an analysis relation to determine average system delay T for given values of λ, μ, B, and τ, Eq. (10) can also be applied to specify any one of these five parameters in terms of the other four. An important case among such possibilities is that of determining what τ must be for given λ, μ, and B in order to satisfy a specification on T. In this case, we are determining the speed at which the processor must be capable of forming packets. The corresponding determination of τ is one of solving the quadratic algebraic equation

$$\lambda\tau^2 - 2(1 + \lambda T)\tau + 2T(1 - e^{-\mu B}) = 0$$

to obtain

$$\tau = \frac{(1 + \lambda T) \pm \sqrt{1 + 2\lambda T e^{-\mu B} + \lambda^2 T^2}}{\lambda} \qquad (11)$$

Since both solutions are positive, we might be tempted to select the larger one which calls for a slower, and therefore, less expensive processor. However, satisfaction of the stability condition, Eq. (8),

$$\tau < \frac{1 - e^{-\mu B}}{\lambda}$$

requires that we accept the smaller value for τ.

This concludes our development of the packet formation process model and its performance relations. We summarize by noting that the process parameters are λ, μ, B, and τ; and that the key performance relations are Eqs. (4), (5), and (7) through (11).

3. EXAMPLE

We consider the case where $\lambda = 10$ messages/sec, $1/\mu = 1000$ bits, $B = 1000$ bits, and $\tau = 50$ msec. From Eq. (7), $\rho = 0.79$ so the process is stable.

Successively, from Eqs. (4), (5), (9), and (10), we get

$\bar{y} = 79.1$ msec

$\bar{n} = 1.58$ packets/message

$W = 204.2$ msec

$T = 283.3$ msec

We note that the average queueing time is more than 2.5 times the average service time. Moreover, since the specification of average source to destination node delay is on the order of 200 msec for interactive applications in packet-switched networks, we see that packet formation introduces an inordinate delay in this case.

If we agree that T = 50 msec is acceptable, we obtain from Eq. (11) the result that $\tau = 22.8$ msec is required. Under this requirement, we also get

$\bar{y} = 36.1$ msec

$\bar{n} = 1.58$ packets/message

$\rho = 0.36$

$W = 13.9$ msec

To go even further and make T negligible relative to 200 msec, say $T = 2$ msec, requires $\tau = 1.24$ msec which is a physically realistic processor capability. In this instance, the other variables which characterize the packet formation process are $\bar{y} = 1.97$ msec, $\bar{n} = 1.58$ packets/message, $\rho = 0.0197$, and $W = 0.03$ msec. Clearly, the price paid to achieve this rapid response is that of an extremely low average utilization.

4. CONCLUSION

In this paper, we have developed and illustrated an M/G/1 queueing system model for the packet formation process in packet-switched telecommunication netorks. The model is defined by the four process parameters: λ = average message arrival rate, $1/\mu$ = average message length, B = maximum length of text field in a packet, and τ = processor time to compose one packet. All of these are physically measurable variables; λ and $1/\mu$ are generally user imposed; and B and τ are design variables.

Implicitly, the treatment here has been for information packets, but it should also be applicable to control packets. A number of aspects of the packet formation process have not, however, been treated and are suggested for future research. Among these are characterization of the output process, i.e., the packet flow in Fig. 1, the modeling of the process for messages with priorities, the possibility of parallel processing, e.g., multiple servers, and the effects of errors in packet formation.

REFERENCES

1. L. Kleinrock, Queueing Systems: Vol. II, Computer Applications. New York: Wiley-Interscience, 1976.

2. M. Schwartz, Computer-Communication Network Design and Analysis. Englewood Cliffs, N.J.: Prentice-Hall, 1977.

"REGULARITY" AND DELAY IN QUEUEING SYSTEMS

A. EPHREMIDES
University of Maryland
College Park, Maryland 20742

In this contribution we consider the "folk theorem" that states that the more "regular" the customer arrival patterns and/or the service time patterns are, the lower the value of the average waiting time per customer in a queueing system.

This problem is of interest in computer communication networks where precautions can be taken in the design process to make message lengths (that correspond to service times) and message interarrival times (by appropriate multiplexing) less "variable" thereby reducing the average message delay.

We try to establish precise statements of versions of this folk theorem. In particular we consider standard M/M/1, M/G/1, G/M/1, and some G/G/1 systems and establish the validity of this "theorem". Furthermore we consider a problem of routing, where the means by which the traffic is split can influence (through the folk theorem) the value of the delay. In particular Bernoulli splitting is shown to be generally inferior to more "deterministic" means of splitting the traffic.

A QUEUEING MODEL FOR ANALYSIS OF A BURSTY MULTIPLE-ACCESS COMMUNICATION CHANNEL

J.W. SENNOTT and L.I. SENNOTT
Dept. of Elec. Engr. and Elec. Engr. Techn., Bradley Univ.,Peoria,IL 61625 and Dept. of Math, Illinois State Univ., Normal, IL 61761

We consider a bursty multiple-access communication channel with constrained total system bandwidth, total average power, and message error rates. The somewhat novel aspect of this paper is its attempt at uniting a queueing model for bursty sources with a multiple-access coding theorem for steady sources. This treatment differs from much current work on systems such as ALOHA in three respects: 1) messages are not packetized, 2) message queueing is allowed at each transmitter, and 3) a multiple-access coding theorem is employed, so that messages overlapping in time will not conflict and retransmission due to collision is never required.

Because of the bursty nature of the sources, it is expected that most of the time, many of the transmitters will be idle. This indicates the feasibility of a dynamic allotment of transmitter power and/or bandwidth. Let m(t) be a random process representing the number of transmitters active at time t. By means of an approximation, m(t) is modeled as a birth-death process. The approximation appears good in both the light and heavy traffic cases.

To develop the message service rates needed in the queueing model, a multiple-access coding theorem is utilized to construct four schemes for the assignment of power and coding rate to active transmitters. These schemes are compared under the expected message system time criterion.

Necessary and sufficient conditions on the parameters for system operation are given, and all schemes are shown to have the same saturation behavior. The adaptive coding schemes are shown to give much reduced expected message system time over the fixed coding schemes except near saturation.

ON DESIGN OF THE LEAST VULNERABLE NETWORKS OF MINIMUM DELAY

H. LEE
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

ABSTRACT

In this paper, we consider the problem of interconnecting a large number of modules or processing elements in current digital systems. The routing delay and the reliability or survivability of the network are modeled as the diameter and the node-connectivity of the corresponding network graph. We present an algorithm to construct a family of regular, directed graphs and show that the graph satisfies both of the minimum diameter and the maximum connectivity criteria.

Keywords: diameter, node-connectivity, directed graph

1. Introduction

We consider the problem of interconnecting a large number of modules or processing elements in current digital systems. A digital and/or computer communication network is modeled by a linear, directed graph G = (V,E), where V is the set of nodes and E is the set of edges in G. Each node represents a module or processing element and each edge corresponds to a communication link in the network. The indegree and outdegree of a node denote the number of edges directed into and from that node respectively. A regular, directed graph is a graph in which all the nodes have the same indegree and outdegree. There are two points of concern: the maximum routing delay, and the reliability or survivability of the network. They are modeled as the diameter and the node-connectivity of the corresponding network graph.

We denote the length of a shortest path from node i to j in G by d_{ij}, where the length of a path is determined by the number of edges in the path. Then the diameter of G is defined by [9]:

$$k(G) = \max_{i,j \in V} d_{ij}$$

Therefore the maximum time to route a message through the network increases with the diameter of the corresponding network graph and it is desirable to have a network graph with a small diameter when given the number of nodes and degree of the graph.

We denote the number of nodes in a regular, directed graph with degree d and diameter k as n(d,k). It can be easily seen that

$$n(d,k) \leqq 1 + d + d^2 + \dots + d^k$$

$$= \frac{d^{k+1} - 1}{d-1} \quad (d > 1) \qquad (1)$$

A (d,k) directed graph which achieves the above bound is called a (d,k) directed Moore graph in [3]. However, it has been shown [3] that no (d,k) directed Moore graphs exist for $d > 1$ and $k > 1$.[1] In the search for regular networks that come as close as possible to the bound of (1), we pose the problem in a different functional form. That is, given the number of nodes n and degree d, construct a network with the minimal diameter. From (1) we have

$$k(n,d) \geqq \lceil \log_d \{n(d-1) + 1\} \rceil - 1$$

$$= k_{min}(n,d) \qquad (2)$$

for given n and d.

In $G = (V,E)$ we define a number ρ_{ij} for each pair of nodes i and j as the number of node-disjoint paths between nodes i and j. Then the node-connectivity of G is given by [5,9]:

$$c(G) = \min_{i,j \in V} \rho_{ij}$$

Note that the node-connectivity (from now on, we simply use the term "connectivity" to denote the node-connectivity) of a graph cannot exceed the degree of the graph. Hence, in a regular, directed graph G of degree d

$$c(G) \leqq d \qquad (3)$$

The connectivity of a network graph is one direct measure of the reliability or survivability of the network [11,13]. It is necessary to remove at least c(G) nodes to isolate at least one pair of the remaining nodes or, equivalently, there still exists at least one path between any pair of nonfaulty nodes in case of up to $c(G) - 1$ node failures in the network. Some construction techniques to find a graph with the maximum connectivity are given in [8,13,14]. However, the diameter of the graphs considered in [8.13,14] increases rather linearly with the number of nodes in the graph or is far from the minimum value.

In this paper, we present an algorithm to construct a regular, directed graph of degree d and show that it achieves the minimum diameter as well as the maximum connectivity.

---

[1] Similar study on undirected Moore graphs also shows [2] that only a few ((3,2), (7,2) and possibly (57,2)) (d,k) Moore graphs are realizable. The problem of maximizing the number of nodes n given degree d and diameter k or minimizing the diameter k of an undirected graph of degree d with n nodes were considered in [1,4,7,12].

2. Construction of $G_{opt} = (V,E)$

For given degree d of the graph, we assume that the number of nodes $n = d^x$ for some positive integer x. We identify a node i $(0 \leqq i < n)$ by x d-ary digits, i.e., $i \equiv i_0 i_1 \ldots i_{x-2} i_{x-1}$ where $0 \leqq i_\alpha < d$ for $\alpha = 0,1,\ldots,x-1$. We call i_α the $(\alpha + 1)$th bit of node i. We denote the set of nodes all of whose bits are of the same by V_s. More precisely

$$V_s = \{i_0 i_1 \ldots i_{x-2} i_{x-1} \mid i_0 = i_1 = \ldots = i_{x-2} = i_{x-1} = \ell \text{ for } 0 \leqq \ell < d\}$$

We construct the graph $G_{opt} = (V,E)$ as follows:

Step 1. For node $i \in V - V_s$, place an edge from node i $(\equiv i_0 i_1 \ldots i_{x-2} i_{x-1})$ to j for $j \equiv i_1 i_2 \ldots i_{x-1} *$, where $* = 0,1,\ldots,d-2,\ d-1$.

Step 2. For node $i \in V_s$, let $i_0 = i_1 = \ldots = i_{x-2} = i_{x-1} = \ell\,(0 \leqq \ell < d)$. Place an edge from node i to j for $j \equiv i_1 i_2 \ldots i_{x-1} *_{(-\ell)}$, where $*_{(-\ell)} = 0,1,\ldots,\ell-1,\ell+1,\ldots,d-2,\ d-1$, and to $j' \equiv j'_0 j'_1 \ldots j'_{x-2} j'_{x-1}$, where

$$j'_\alpha\ (0 \leqq \alpha < x) = \begin{cases} (\ell + 1) & \text{if } 0 \leqq \ell < d-1 \\ 0 & \text{if } \ell = d-1 \end{cases}$$

3. Theorems

In this section, we show that $G_{opt} = (V,E)$ constructed by the algorithm in section 2 achieves both the minimum diameter and the maximum connectivity.

Theorem 1: Let $G_{opt} = (V,E)$ be a regular, directed graph with $n(= d^x)$ nodes and degree d, constructed by the algorithm in section 2. Then i) the diameter $k(G_{opt}) = x$; and ii) $k(G_{opt}) = k_{min}(n,d)$.

Proof: i) Given an arbitrary pair of nodes i and j $(0 \leqq i, j < n)$ node j can be reached from node i in at most x steps, i.e., a path:

$$i_0 i_1 \ldots i_{x-2} i_{x-1} (\equiv i) \rightarrow i_1 i_2 \ldots i_{x-1} j_0 \rightarrow i_2 i_3 \ldots i_{x-1} j_0 j_1 \rightarrow \ldots \rightarrow$$
$$i_{x-2} i_{x-1} j_0 \ldots j_{x-4} j_{x-3} \rightarrow i_{x-1} j_0 \ldots j_{x-3} j_{x-2} \rightarrow j_0 j_1 \ldots j_{x-2} j_{x-1} (\equiv j)$$

will do. Now, we note that the distance from node 00 - 00 $(\equiv 0)$ to node 10 - 00 $(\equiv \frac{n}{d})$ is no less than x. Hence, the diameter of G_{opt} is equal to x. ii) By substituting d^x for n in eq. (2), it can be readily seen that $k_{min}(n,d) = x$. Q.E.D.

To show the maximum connectivity of G_{opt} we use the following algorithm due to S. Even [6] which tests whether the connectivity of a directed graph $G = (V,E)$ is at least s.

S. Even's Algorithm[2]

1. For every node i and j such that $0 \leqq i < j < s$, check whether there are s node-disjoint paths from node i to j, and vice versa. If one of these tests fails, then the connectivity of G is less than s.
2. For every node j, $s \leqq j < n$, form an augmented graph $\overrightarrow{G}_j$ constructed from G by adding a new node a and placing an edge from node a to each of the nodes in $L = \{0,1...,j-1\}$. Check whether there are s node-disjoint paths from node a to j. Similarly form $\overleftarrow{G}_j$ by adding a new node a and placing an edge from each of the nodes in $L = \{0,1,..., j-1\}$ to node a, and check whether there are s node-disjoint paths from node j to a. If for some j one of these tests fails, then the connectivity of G is less than s.
3. The connectivity of G is at least s. ☐

Now we are ready to state the following theorem.

Theorem 2: Let $G_{opt} = (V,E)$ be a regular, directed graph with n $(= d^x)$ nodes and degree d, constructed by the algorithm in section 2. Then G_{opt} has the maximum connectivity, i.e., $c(G_{opt}) = d$.

Proof: The following proof is essentially applying S. Even's algorithm, with $s = d$, to G_{opt}. Let $P^0_{ij}, P^1_{ij}, \ldots, P^{(d-1)}_{ij}$ denote d node-disjoint paths from node i to j for $0 \leqq i, j < d$ in G_{opt}. We construct the paths P^t_{ij} $(0 \leqq t < d)$, except for i) $i = 0$ and $t = j$ and ii) $i \neq 0$ and $t = i$, as follows:

P^0_{ij}	$P^t_{ij} (0 < t < d)$
00 ------ 0i (≡ i)	00 ------ 0i (≡ i)
00 ----- 0i0	00 ----- 0it
00 ---- 0i00	00 ---- 0it0
⋮	⋮
0i0 ----- 00	0it0 ---- 00
i0 ------ 00	it0 ----- 00
00 ------ 0j (≡ j)	t00 ----- 00
	00 ------ 0j (≡ j)

For the cases of i) $i = 0$ and $t = j$, and ii) $i \neq 0$ and $t = i$, we construct P^j_{0j} $(0 < j < d)$ and P^i_{ij} $(i \neq 0)$ as follows:

---

[2] Those who are interested in the proof of validity of the algorithm are referred to [5,6].

P_{0j}^{j} $(0 < j < d)$	P_{ij}^{i} $(i \neq 0)$
$00 \text{ ------ } 00$ $(\equiv 0)$	$00 \text{ ------ } 0i$ $(\equiv i)$
$11 \text{ ------ } 11$	$00 \text{ ----- } 0ii$
$22 \text{ ------ } 22$	$00 \text{ ---- } 0iii$
$\vdots$	$\vdots$
$jj \text{ ------ } jj$	$0i \text{ ------ } ii$
$jj \text{ ------ } j0$	$ii \text{ ------ } ii$
$jj \text{ ----- } j00$	$(i+1)(i+1) \text{ ------ } (i+1)(i+1)$
$\vdots$	$\vdots$
$jj0 \text{ ----- } 00$	$(d-1)(d-1) \text{ ------ } (d-1)(d-1)$
$j0 \text{ ------ } 00$	$00 \text{ ------ } 00$
$00 \text{ ------ } 0j$ $(\equiv j)$	$00 \text{ ------ } 0j$ $(\equiv j)$

It is easy to see that the d paths P_{ij}^{t}, $0 \leqq t < d$ and $0 \leqq i, j < d$, constructed as above are node-disjoint.

For every node j, $d \leqq j < n$, we construct $\vec{G}_{opt,j}$ by adding a new node a and placing an edge from node a to each of the nodes in $L = \{0,1,\ldots,j-1\}$. We denote the set of d node-disjoint paths from node a to j in $\vec{G}_{opt,j}$ by $\{P_{aj}^{t}\}$, $0 \leqq t < d$. Let

$$V_\alpha = \{\overbrace{00 \text{—} 0}^{\theta}\,\overbrace{\ell\ell \text{——} \ell}^{x-\theta-1} j_{x-1} \mid 0 \leqq \theta \leqq x-2,\ 2 \leqq \ell \leqq d-1 \text{ and } j_{x-1} = 0,1,\ldots,d-1\}.$$

If $j \notin V_\alpha$, then we construct P_{aj}^{t}, $0 \leqq t < d$, as follows:

P_{aj}^{0}	P_{aj}^{1}	P_{aj}^{t} $(2 \leqq t < d)$
a	a	a
$0j_0j_1 \cdots j_{x-3}j_{x-2}$	$00 \text{ — } 00$	$00 \text{ — } 0t$
$j_0j_1 \cdots j_{x-2}j_{x-1}$ $(\equiv j)$	$11 \text{ — } 11$	$00 \text{ — } 0tt$
	$11 \text{ — } 1j_0$	$\vdots$
	$\vdots$	$0t \text{ — } tt$
	$1j_0j_1 \cdots j_{x-3}j_{x-2}$	$tt \text{ — } tj_0$
	$j_0j_1 \cdots j_{x-2}j_{x-1}$ $(\equiv j)$	$tt \text{ — } tj_0j_1$
		$\vdots$
		$tj_0j_1 \cdots j_{x-3}j_{x-2}$
		$j_0j_1 \cdots j_{x-2}j_{x-1}$ $(\equiv j)$

If $j \in V_\alpha$ and $\theta \neq 0$, then $j \equiv j_0j_1\cdots j_{x-2}j_{x-1}$ where $j_0 = 0$, $j_1 = j_2 = \cdots = j_{\theta-1} = 0$ and $j_\theta = j_{\theta+1} = \cdots = j_{x-2} = \ell$ for some θ and ℓ; $2 \leqq \theta \leqq x-2$ and $2 \leqq \ell \leqq d-1$. Hence we note that P^{t}_{aj}, $0 \leqq t < d$, constructed as above are not node-disjoint. More precisely P^{0}_{aj} and P^{ℓ}_{aj} share a common intermediate node $\overbrace{00 \text{ --- } 0}^{\theta}\overbrace{\ell\ell \text{ --- } \ell}^{x-\theta-1}$. We construct the following two paths,

$$\begin{array}{l} P^{1'}_{aj} \\ \hline \quad a \\ 00 \text{ --- } 01j_0j_1\cdots j_{x-2-\theta} \\ 00 \text{ --- } 01j_0j_1\cdots j_{x-1-\theta} \\ \qquad \vdots \\ 01j_0j_1\cdots j_{x-4}\,j_{x-3} \\ 1j_0j_1\cdots j_{x-3}\,j_{x-2} \\ j_0j_1\cdots j_{x-2}\,j_{x-1}(\equiv j) \end{array}$$

$$\begin{array}{l} P^{\ell'}_{aj} \\ \hline \quad a \\ 00 \text{ --- } 00 \\ 11 \text{ --- } 11 \\ 11 \text{ --- } 1\ell \\ \qquad \vdots \\ 1\ell \text{ --- } \ell\ell \\ \ell\ell \text{ --- } \ell j_0 \\ \ell\ell \text{ --- } \ell j_0j_1 \\ \qquad \vdots \\ \ell j_0j_1\cdots j_{x-3}\,j_{x-2} \\ j_0j_1\cdots j_{x-2}\,j_{x-1}(\equiv j) \end{array}$$

Then it can be easily seen that, for $j \in V_\alpha$ and $\theta \neq 0$, $P^{t'}_{aj}$, $0 \leqq t < d$, are node-disjoint, where $P^{t'}_{aj} = P^{t}_{aj}$ for $t \neq 1, \ell$, and $P^{1'}_{aj}$ and $P^{\ell'}_{aj}$ are as above.

If $j \in V_\alpha$ and $\theta = 0$, then $j \equiv \overbrace{\ell\ell \text{ --- } \ell}^{x-1}j_{x-1}$ for $2 \leqq \ell \leqq d-1$ and $j_{x-1} = 0,1,\ldots,d-1$. Then we construct d node-disjoint paths $P^{t''}_{aj}$, $0 \leqq t < d$, as follows:

$$\begin{array}{l} P^{t''}_{aj} \ (0 \leqq t < d,\ t \neq \ell) \\ \hline \quad a \\ 0t\ell \text{ --- } \ell\ell \\ t\ell\ell \text{ --- } \ell\ell \\ \ell\ell \text{ --- } \ell j_{x-1}(\equiv j) \end{array}$$

$$\begin{array}{l} P^{\ell''}_{aj} \\ \hline \quad a \\ (\ell-1)(\ell-1) \text{ --- } (\ell-1)(\ell-1) \\ \ell\ell \text{ --- } \ell\ell \\ \ell\ell \text{ --- } \ell j_{x-1}\ (\equiv j) \end{array}$$

Now for every node j, $d \leqq j < n$, we construct $\overleftarrow{G}_{opt,j}$ by adding a new node a and placing an edge from each of the nodes in $L = \{0,1,\ldots,j-1\}$ to node a. We denote the set of d node-disjoint paths from node j to a in $\overleftarrow{G}_{opt,j}$ by $\{P^{t}_{ja}\}$, $0 \leqq t < d$. Let

$$V_\beta = \{j_0\overbrace{\ell\ell \text{———} \ell}^{\gamma}\overbrace{00 \text{———} 0}^{x-\gamma-1} \mid 0 \leqq \gamma \leqq x-1,\ 1 \leqq \ell \leqq d-2 \text{ and } j_0 = 0,1,\ldots,d-1\}.$$

If $j \notin V_\beta$, then we construct P_{ja}^{t}, $0 \leqq t < d$, as follows:

P_{ja}^{0}

$j_0j_1 \cdots\cdots\cdots j_{x-2}\ j_{x-1} (\equiv j)$
$j_1j_2 \cdots\cdots\cdots j_{x-1}(d-1)$
$j_2j_3 \cdots\ j_{x-1}(d-1)(d-1)$
$\vdots$
$j_{x-1}(d-1) \text{ —— } (d-1)(d-1)$
$(d-1)(d-1) \text{ —— } (d-1)(d-1)$
$0\ 0 \underset{a}{\text{————————}} 0\ 0$

$P_{ja}^{t}(1 \leqq t \leqq d-2)$

$j_0j_1 \cdots j_{x-2}\ j_{x-1}\ (\equiv j)$
$j_1j_2 \cdots j_{x-1}\ t$
$j_2j_3 \cdots j_{x-1}\ tt$
$\vdots$
$j_{x-1}\ t \text{ ———— } tt$
$tt \text{ ———————— } t0$
$tt \text{ ———————— } t00$
$\vdots$
$t0 \text{ ———————— } 00$
$00 \underset{a}{\text{————————}} 0t$

$P_{ja}^{(d-1)}$

$j_0j_1 \cdots\cdots\cdots j_{x-2}\ j_{x-1} (\equiv j)$
$j_1j_2 \cdots\cdots\cdots j_{x-1}\ 0$
$j_2j_3 \cdots\cdots j_{x-1}\ 0\ (d-1)$
$j_3j_4 \cdots j_{x-1}\ 0\ (d-1)\ 0$
$\vdots$
$j_{x-1}\ 0\ (d-1)\ 0 \text{ ——— } 00$
$0\ (d-1)\ 0 \text{ ———————— } 00$
$(d-1)\ 0 \text{ ———————— } 00$
$00 \underset{a}{\text{————————}} 0(d-1)$

If $j \in V_\beta$ and $\gamma \neq 0$ or $x-1$, then $j \equiv j_0j_1\cdots j_{x-2}\ j_{x-1}$ where $j_1 = j_2 = \cdots = j_\gamma = \ell$ and $j_{\gamma+1} = j_{\gamma+2} = \cdots = j_{x-1} = 0$ for some γ and ℓ; $1 \leqq \gamma \leqq x-2$ and $1 \leqq \ell \leqq d-2$. We note that P_{ja}^{ℓ} and $P_{ja}^{(d-1)}$ share a common intermediate node $\overbrace{\ell\ell \text{———} \ell}^{\gamma}\overbrace{00 \text{———} 0}^{x-\gamma}$. We construct $P_{ja}^{\ell'}$ as follows:

$P^{\ell'}_{ja}$

$j_0 j_1 \cdots j_{x-2}\, j_{x-1}$ $(\equiv j)$
$j_1 j_2 \cdots j_{x-1}\, \ell$
$j_2 j_3 \cdots j_{x-1}\, \ell\, 0$
$\vdots$
$j_{x-1}\, \ell\, 0 \text{ ——— } 00$
$\ell\, 00 \text{ ——— } 00$
$00 \underset{a}{\text{ ——— }} 0\ell$

Then $P^{t'}_{ja}$, $0 \leqq t < d$, where $P^{t'}_{ja} = P^{t}_{ja}$ for $0 \leqq t < d$ and $t \neq \ell$, and $P^{\ell'}_{ja}$ is as above, ard d node-disjoint paths from node j to a. If $j \in V_\beta$ and $\gamma = 0$, then $j \equiv j_0 \overbrace{0 \text{ ——— } 00}^{x-1}$ for $1 \leqq j_0 < d$. Then we construct d node-disjoint paths $P^{t''}_{ja}$, $0 \leqq t < d$, as follows;

$P^{t''}_{ja}$ $(0 \leqq t < d)$

$j_0 0 \text{ ——— } 00$ $(\equiv j)$
$00 \underset{a}{\text{ ——— }} 0t$

If $j \in V_\beta$ and $\gamma = x - 1$, then $j \equiv j_0 \overbrace{\ell \text{ ——— } \ell\ell}^{x-1}$ for $0 \leqq j_0 < d$ and $1 \leqq \ell \leqq d-2$. Then we construct d node-disjoint paths $P^{t'''}_{ja}$, $0 \leqq t < d$, as follows;

$P^{t'''}_{ja}$ $(0 \leqq t < d,\ t \neq 0 \text{ or } \ell)$

$j_0\, \ell \text{ ——— } \ell\ell$ $(\equiv j)$
$\ell\ell \text{ ——— } \ell t$
$\ell\ell \text{ ——— } \ell t 0$
$\vdots$
$\ell t 0 \text{ ——— } 00$
$t0 \text{ ——— } 00$
$00 \underset{a}{\text{ ——— }} 0t$

$P^{0'''}_{ja}$

$j_0\, \ell \text{ ——— } \ell\ell$ $(\equiv j)$
$\ell\ell \text{ ——— } \ell\ell$
$(\ell+1)(\ell+1) \text{ ——— } (\ell+1)(\ell+1)$
$\vdots$
$(d-1)(d-1) \text{ ——— } (d-1)(d-1)$
$00 \underset{a}{\text{ ——— }} 00$

$$\begin{array}{lcl}
\multicolumn{3}{l}{P_{ja}^{\ell'''}} \\
\hline
j_0\ell & \text{——} & \ell\ell \ (\equiv j) \\
\ell\ell & \text{——} & \ell 0 \\
\ell\ell & \text{——} & \ell 00 \\
 & \vdots & \\
\ell 0 & \text{——} & 00 \\
00 & \underset{a}{\text{——}} & 0\ell
\end{array}$$

This completes the proof. Q.E.D.

4. Discussion

We have shown that a family of regular, directed graphs obtained by the presented construction achieves both minimum diameter and maximum connectivity. An immediate extension to the presented result is to allow the number of nodes, n to be arbitrary. Then we have a generalized construction in which node i is connected to node j such that $j = i\cdot d + *$ (modulo n), $* = 0,1,\ldots,d-1$, which replaces step 1 of the presented construction. Step 2 of the presented construction can be accordingly modified. A recent result by Imase and Ito [10] shows that the generalized construction achieves diameter very close to the minimum value, i.e., the diameter is shown to be equal to or greater than $k_{\min}$ by at most one. However, it still remains as an interesting question whether the generalized construction also achieves the maximum connectivity. Another point of concern is to address to the undirected graph case. That is, if we make each edge undirected in the presented or generalized construction, we want to see how close the connectivity of the graph can be realized to the maximum value.

Acknowledgement

The author would like to thank D. K. Pradhan and S. L. Hakimi for interesting discussions and comments.

References

[1] B. Arden and H. Lee, "A Multi-Tree-Structured Network", Proc. COMPCON 78, pp. 201-210, Sept. 1978.

[2] E. Bannai and T. Ito, "On finite Moore graphs", J. Fac. Sci., Tokyo Univ., pp. 191-208, 1973.

[3] W. G. Bridges and S. Toueg, "On the impossibility of directed Moore graphs", J. Combinatorial Theory (B), to appear.

[4] B. Elspas, "Topological constraints on interconnection limited logic", Switching Theory and Logic Design, Vol. S-164, pp. 133-147, Oct. 1964.

[5] S. Even, Graph Algorithms, Computer Science Press, 1979.

[6] ________, "An algorithm for determining whether the connectivity of a graph is at least k", SIAM J. Comput., Vol. 4, No. 3, pp. 393-396, Sept. 1975.

[7] H. Friedman, "A design for (d,k) graphs", IEEE Trans. Elec. Comp., Vol. EC-15, pp. 253-254, Apr. 1966.

[8] S. L. Hakimi, "An algorithm for construction of the least vulnerable communication network or the graph with the maximum connectivity", IEEE Trans. Circuit Theory (Corresp.), Vol. CT-16, pp. 229-230, May 1969.

[9] F. Harary, Graph Theory, Addison-Wesley, 1969.

[10] M. Imase and M. Ito, "Design to minimize diameter on building-block network", IEEE Trans. Comp., to appear.

[11] I. Rubin, "On reliable topological structures for message-switching communication networks", IEEE Trans. Comm., Vol. COM-26, No. 1, pp. 62-74, Jan. 1978.

[12] R. M. Storwick, "Improved construction technique for (d,k) graphs", IEEE Trans. Elec. Comp., pp. 1214-1216, Dec. 1970.

[13] R. S. Wilkov, "Analysis and design of reliable computer networks", IEEE Trans. on Comm., Vol. COM-20, pp. 660-678, June 1972.

[14] ____________, "On maximally connected graph of minimal diameter", Proc. 1970 IEEE Int'l Symp. on Circuit Theory, pp. 25-26, Dec. 1970.

PERFORMANCE SIMULATION OF RECEIVER FILTERS FOR SERIAL DETECTION OF MSK SIGNALS

JOHN M. LIEBETREU
General Motors Corp.
Flint, Michigan

CARL R. RYAN
Motorola, Inc. - GED
Gilbert, Arizona

ABSTRACT

This paper describes a simplified minimum (frequency) shift keying (MSK) signaling technique, including a means for quadrature detection of a serially generated MSK signal. A computer assisted performance simulation is introduced to acquire performance data for use in quantifying the quadrature detection method of the serial (SMSK) approach. The results verify the similarity between SMSK and coherent biphase shift keying (BPSK). Performance using a conventional quadrature hybrid followed by a three section low-pass filter resulted in a bit error rate degradation of 0.4 dB.

INTRODUCTION

The continually increasing demand for digital transmission channels in the RF band has led many researchers to consider more spectrally efficient modulation techniques to alleviate the problem of spectral congestion. Several authors have reported on the advantages of the minimum (frequency) shift keying (MSK) format over other digital modulation techniques [1]-[4]. In 1977, Amoroso and Kivett [5] presented a simplified MSK signaling technique.

This paper presents detailed performance characteristics of a demodulation technique for the simplified MSK signal using conventional filters in conjunction with a quadrature hybrid to approximate the ideal filters. This technique has special significance for application to high frequency communications systems[6]. Simulated performance data are generated for comparison of these approximations to the theoretical MSK performance.

PERFORMANCE SIMULATION

Analysis of the SMSK System

The block diagram of Figure 1 shows the system model for the performance simulation, which assumes a non-bandlimited channel and an ideal transmitter conversion filter. The simulation is accomplished at baseband, which involves translation of the transmitter conversion filter to baseband to determine its low-pass equivalent.

Assuming carrier regeneration at the receiver with static phase error ϕ, the receiver is modeled as shown in Figure 2, excluding

Filter Two.

Filter Two is composed of a cascade of three single-pole low-pass stages to provide an approximation to a Gaussian low-pass filter [7]. Knowing the frequency spectrum of the BPSK input to the transmitter and the transfer function of each of the system components, the frequency spectrum at the output of Filter Two is available, which is then manipulated by a fast Fourier transform algorithm to yield the time domain waveform.

Indirect BER Estimation

An indirect method of estimating BER performance is available which accurately estimates the error probability with the advantage of small processing times.

This indirect approach assumes a linear channel and linear system components. One bit is passed through the system in the frequency domain and the fast Fourier transform is used to determine the time domain response to this single pulse of duration T. The time domain waveform is determined for $-2T < t < 8T$, where $t = 0$ is the peak value of the pulse response. Making use of the linearity of the system, superposition can be used to construct a random data sequence at the receiver output. This is accomplished by using a seven stage maximal length PN code generator [8].

A sampling algorithm is then introduced to sample the waveform at intervals of T. This sampling is not bit-by-bit adaptive, but is fixed for the entire data stream, as would be the case with practical bit synchronizers. The contribution of the i^{th} bit to the average probability of error at a given signal-to-noise ratio is obtained from the i^{th} sample value. The average probability of error is then determined for K bits from

$$P_{e_{Avg}} = \frac{\sum_{i=1}^{K} P_{e_i}}{K}$$

This procedure is repeated while iterating the reference value of E_b/N_o, the ratio of the bit energy to noise spectral density, to obtain an error probability curve for a given sample point. The sample point is then varied to generate a series of error probability curves. For each curve, the relative degradation in E_b/N_o at 10^{-6} BER is found by linear interpolation. This data is used to plot relative degradation as a function of timing error in the sampler.

Performance Evaluation Tests

The performance of the serial MSK quadrature phase detector receiver is simulated with several common perturbations. Specifically, the performance sensitivity to the following parameters is examined:

1) Filter Two bandwidth;
2) Bandwidth limiting introduced by the quadrature hybrid;
3) Timing adjustment of the bit synchronizer;
4) Demodulator static phase error.

Typical results are presented for two different quadrature hybrids, with bandwidths between nulls of $\frac{1.5}{T}$ and $\frac{1.0}{T}$, to observe the degradation introduced at the receiver by limiting the spectrum main lobe to one data rate, a common practice in spectrum conservation techniques [9].

Sensitivity to Filter Two Bandwidth

Figure 3 illustrates the sensitivity of the BER performance to Filter Two bandwidth at a constant error probability of 10^{-6}. Figure 4 shows the error probability curves for the ideal matched filter, and the quadrature hybrid approximation.

Bit Synchronizer Timing Accuracy

If the sampling instant of the bit synchronizer is varied about the optimum, the plots of Figure 5 are obtained, which show the results for the matched filter and two approximations to the matched filter.

Demodulator Static Phase Error

A common cause of signal distortion in digital communications systems is demodulator static phase error [10]. The effect of this subcarrier tracking error is observed in Figure 6, in which plots of the sensitivity to tracking error for QPSK, BPSK, ideal SMSK and SMSK with quadrature detection is presented. Note that BPSK and ideal SMSK are equivalent for phase errors of less than twenty degrees [5]. Also, the sensitivity to static phase errors is far more pronounced for QPSK than either ideal SMSK or SMSK with quadrature detection.

SUMMARY

This paper provides detailed performance data for a method of quadrature detection of an SMSK signal. This method is shown to have a BER performance of 0.4 dB from theoretical SMSK at 10^{-6} error probability. Several measures of performance for the quadrature detection approach exhibit the feasibility of this method. With the benefits of hardware simplification, SMSK provides an economical spectrum conservation technique.

REFERENCES

[1] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," IEEE Communications Society Magazine, pp. 14-22, July 1979.

[2] H. R. Mathwich, J. F. Balcewicz, and M. Hecht, "The Effect of Tandem Band and Amplitude Limiting on the E_b/N_0 Performance of Minimum (Frequency) Shift Keying (MSK)," IEEE Transactions on Communications, COM-22, pp. 1525-1540, October 1974.

[3] F. Amoroso, "Pulse and Spectrum Manipulation in the Minimum (Frequency) Shift Keying (MSK) Format," IEEE Transactions on Communications (Corresp.), COM-24, pp. 381-384, March 1976.

[4] M. Razbel and S. Pasupathy, "Spectral Shaping in Minimum Shift Keying (MSK)-Type Signals," IEEE Transactions on Communications (Corresp.), COM-26, pp. 189-195, January 1978.

[5] F. Amoroso and J. A. Kivett, "Simplified MSK Signaling Technique", IEEE Transactions on Communications (Concise Papers), COM-25, pp. 433-441, April 1977.

[6] C. R. Ryan, A. R. Hambley, and D. E. Vogt, "760 Mbit/s Serial MSK Microwave Modem," IEEE Transactions on Communications (Corresp.), COM-28, May 1980.

[7] M. Schwartz, Information Transmission, Modulation and Noise, New York: McGraw-Hill, 1959, pp. 282-291.

[8] S. W. Golomb, Digital Communications with Space Applications, New Jersey: Prentice Hall, 1964.

[9] C. L. Cuccia, "New PCM Techniques Stress Spectrum and Dollar Conservation, Part II," Microwave System News, pp. 37-54, February 1977.

[10] C. R. Ryan, "Design Considerations for a QPSK System," Motorola Technical Memorandum, October 1975.

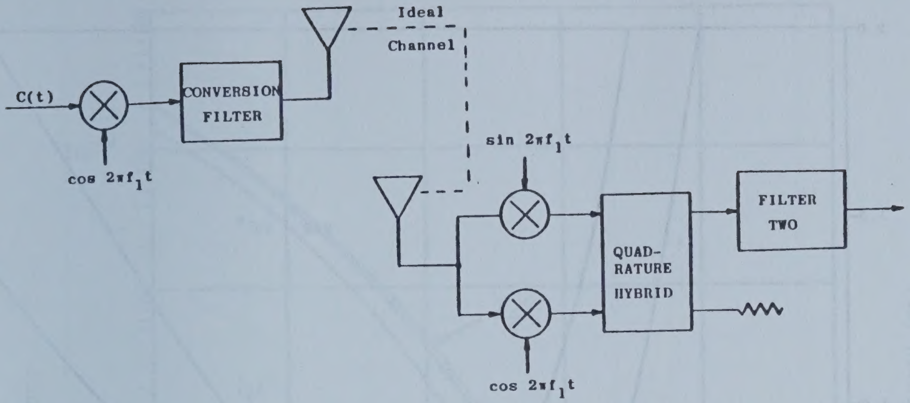

Ideal
Channel
C(t)
CONVERSION FILTER
cos 2πf₁t
sin 2πf₁t
QUAD-RATURE HYBRID
FILTER TWO
cos 2πf₁t

Figure 1 - SMSK SYSTEM

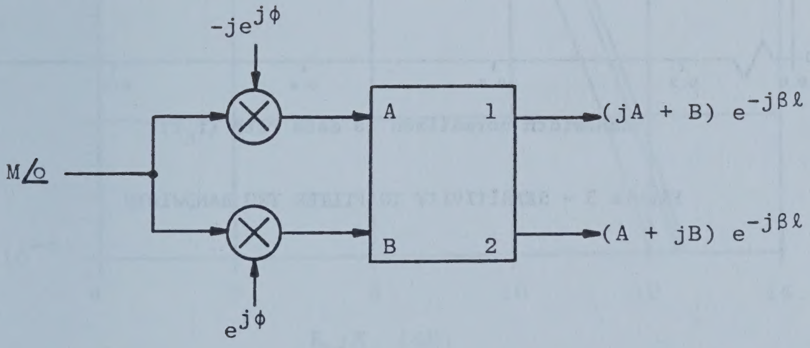

-je^{jφ}
M/0
A
1
B
2
(jA + B) e^{-jβℓ}
(A + jB) e^{-jβℓ}
e^{jφ}

Figure 2 - QUADRATURE HYBRID AT BASEBAND

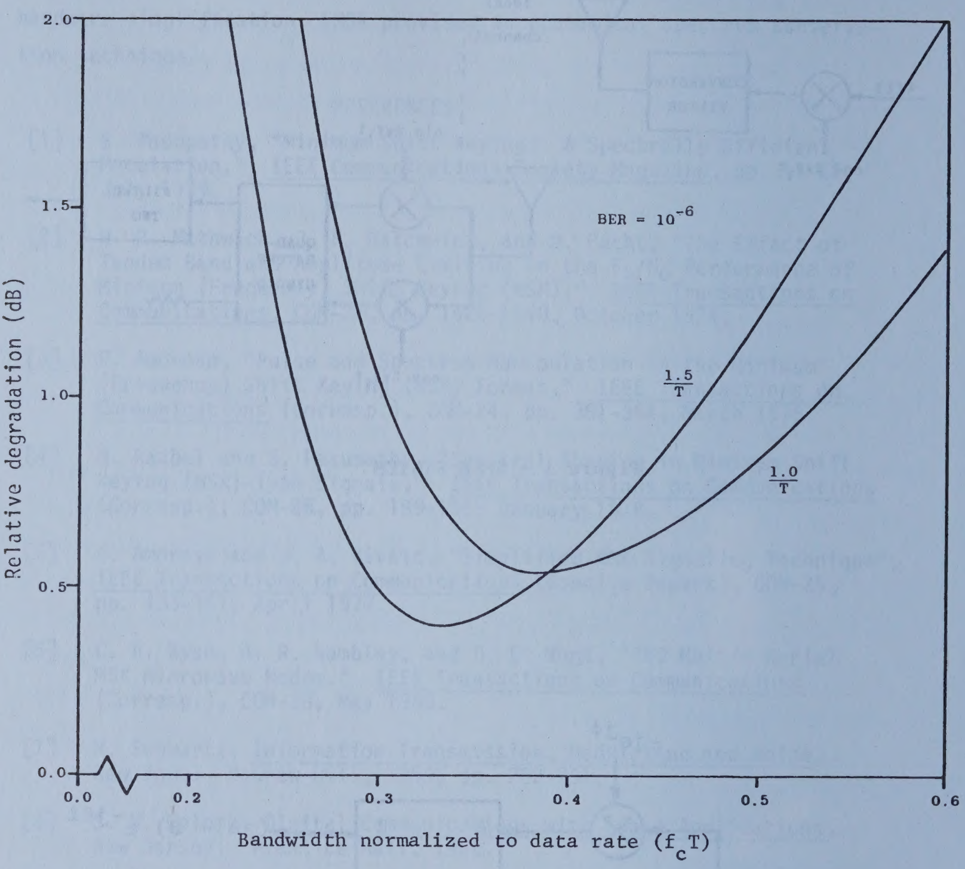


Figure 3 - SENSITIVITY TO FILTER TWO BANDWIDTH

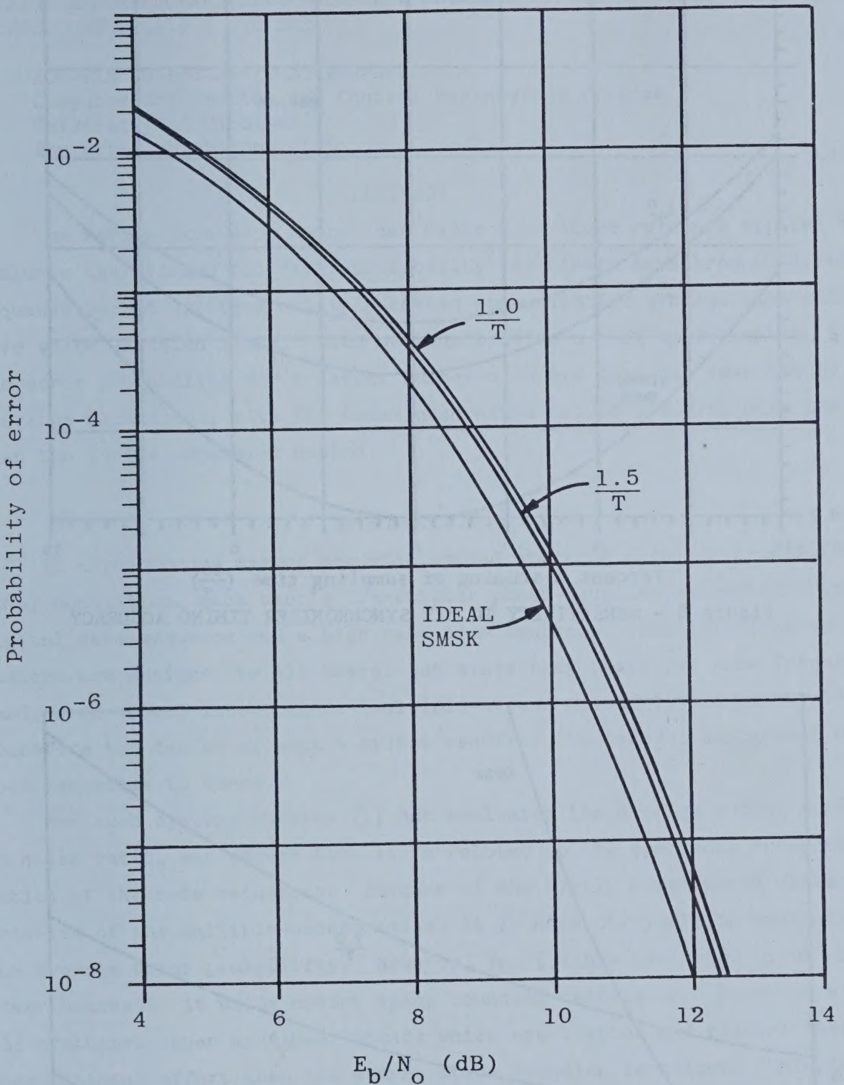


Figure 4 - BER PERFORMANCE

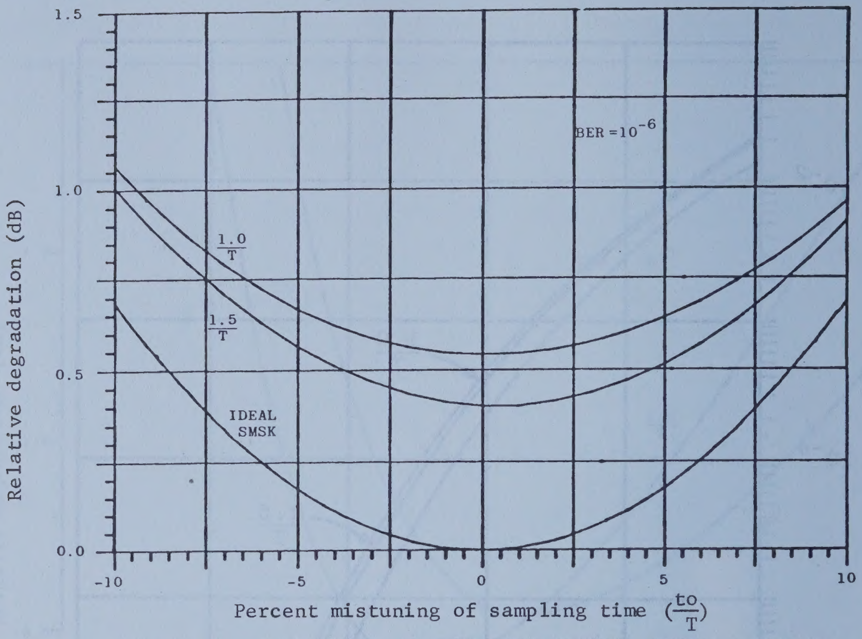


Figure 5 - SENSITIVITY TO BIT SYNCHRONIZER TIMING ACCURACY

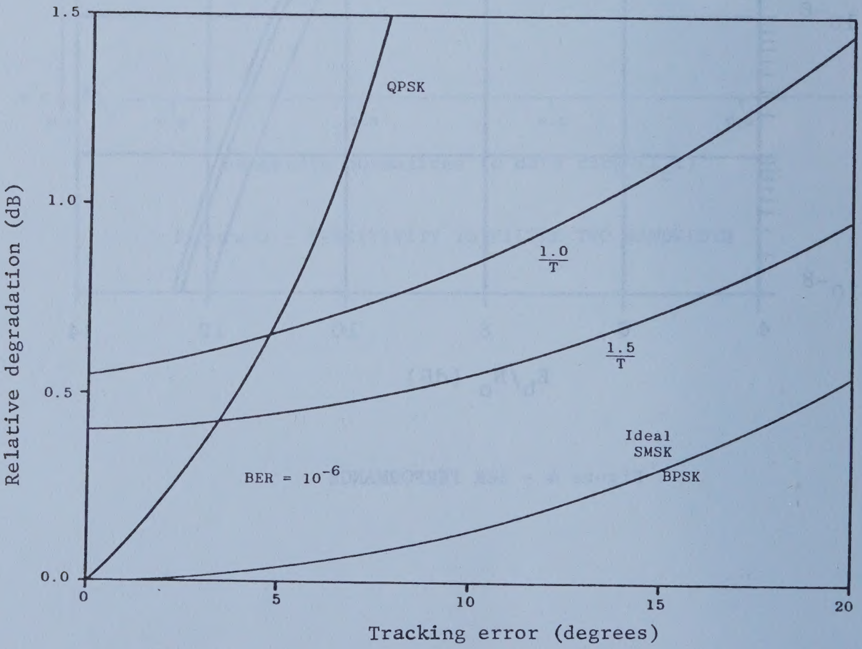


Figure 6 - DEGRADATION DUE TO SUBCARRIER PHASE ERROR

AVERAGE ERROR PROBABILITY FOR DIRECT SEQUENCE SPREAD SPECTRUM MULTIPLE ACCESS COMMUNICATION SYSTEMS

KOU-TAN WU and DAVID L. NEUHOFF
Computer Information and Control Engineering Program
University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

The series expansion method and Gauss quadrature rule are applied to evaluate the average bit error probability for binary asynchronous direct sequence spread spectrum multiple-access communication systems with additive white Gaussian noise. Both methods provide a good approximation to the error probability for a larger number of users than was feasible by previous techniques, with the Gauss quadrature method allowing more users than the series expansion method.

INTRODUCTION

In asynchronous direct sequence spread spectrum multiple-access (DS-SSMA) modulation, each user's transmitter phase-modulates a carrier with a digital data sequence and a high rate code sequence. Distinct code sequences are assigned to all users, but since they share the same frequency band, some mutual interference (multiple-access noise) is unavoidable. Therefore the design of such a system requires the careful assignment of code sequences to users.

For such systems Pursley [1] has evaluated the average output signal to noise ratio, and showed that it is related to the aperiodic autocorrelation of the code sequences. Because of the highly complicated characteristics of the multiple-access noise, it is more difficult to evaluate the average error probability. However, Yao [2] has evaluated upper and lower bounds to it using moment space bounding techniques. Borth et al. [3] evaluated upper and lower bounds which are tighter and require less computational effort than the moment space bounding technique. Unfortunately, both methods are applicable for only a limited number of users. There are two reasons. The complexity of each method increases rapidly with the number of users. And secondly, if the number of users exceeds a threshold depending heavily on the code sequence length (when the maximum normalized multiple-access noise is one or more), then the methods cannot be applied (in the examples we tried, they apply for a maximum of 5 linear m-sequences with code length 127).

The goal of this work was to investigate alternative methods that allow the error probability analysis to be carried out for a larger number of users. Correspondingly, we have applied two techniques from the communication theory literature to this problem: (1) the series expansion method [4,5] and (2) the Gauss quadrature rule [6]. Both techniques have previously been applied to the evaluation of error probability for digital communication systems with intersymbol interference and co-channel interference. Both of these methods require that the moments of the distribution of the multiple-access noise be computed, and both exploit the fact that it is much easier to evaluate the moments than the density of this distribution, especially when the number of users is large.

The series expansion method offers accurate average error probability with moderate complexity. As the number of users increases, the convergence of the series becomes slower, more terms are needed until eventually the method becomes computationally infeasible. The Gauss quadrature rule can also be used to evaluate the average error probability. It works for a larger number of users than the series expansion method. In this paper we describe both methods and test them on several sets of codewords. In addition we compare these results to a calculation based on the assumption that the multiple-access noise is Gaussian.

DS-SSMA SYSTEM

We now describe the model for an asynchronous DS-SSMA system that can be used to simultaneously transmit binary data from K users in a common band with center frequency f_c. Let $b_{k,\ell}$ denote the ℓ-th bit from the k-th user. We assume that data bits are emitted at the rate of one every T seconds and that each $b_{k,\ell}$ is ±1 with equal probability and independent of other $b_{i,j}$'s. Let P be the power available to each user's transmitter. The transmitted signal from the k-th user is

$$s_k(t) = \sqrt{2P}\, b_k(t)\, \tilde{a}_k(t) \cos(2\pi f_c t + \theta_k) \tag{1}$$

where $b_k(t)$ is a rectangular information waveform which equals $b_{k,\ell}$ in the time interval $[\ell T, (\ell+1)T)$, where $\tilde{a}_k(t)$ is a rectangular waveform consisting of periodic repetitions of a code waveform $a_k(t)$ which has duration T seconds and is made up of N rectangular chips, each with duration $T_c = T/N$ seconds and amplitude ±1, and where θ_k is the carrier phase. The code waveform $a_k(t)$ has the form

$$a_k(t) = a_k^{\ell}; \qquad \ell T_c \le t < (\ell+1)T_c \tag{2}$$

for some ±1 valued sequence $a_k^0, a_k^1, \ldots, a_k^{N-1}$. The bandwidth of the signal is proportional to $T_c^{-1}=N/T$, so N can be considered to be a bandwidth expansion factor.

The receiver for the k-th user receives the signal

$$r_k(t) = \sqrt{2P} \sum_{k=1}^{K} \tilde{a}_k(t-\tau_k)b_k(t-\tau_k)\cos(2\pi f_c t+\phi_k) + n_k(t) \tag{3}$$

where $n_k(t)$ is white Gaussian noise that is independent of all other r.v.'s and has two-sided power spectral density $N_o/2$, and where the time delay τ_k and phase $\phi_k \triangleq \theta_k - 2\pi f_c\tau_k$ model the asynchronism between users. We assume that τ_k is uniformly distributed on $[0,T]$, that ϕ_k is uniformly distributed on $[0,2\pi]$, and that τ_k and ϕ_k are independent of each other and all other r.v.'s in this model.

The receiver for the k-th user is assumed to coherently correlate $r_k(t)$ with $a_k(t-\tau_k)\cos(2\pi f_c t+\phi_k)$ over the interval $[\tau_k, T+\tau_k]$ and make a decision about $b_{k,0}$ based on the sign of this correlation. We will concentrate entirely on receiver number one and, without loss of generality, we will assume $\tau_1 = \phi_1 = 0$. Assuming $f_c >> T_c^{-1}$, the output of this correlation is

$$Z_1 = \sqrt{\frac{P}{2}}\, T\, b_{1,0} + \sqrt{\frac{P}{2}}\, u + n \tag{4}$$

where the first term is due to the desired signal, the second term is due to interference from other users, and n is due to the white Gaussian noise. In particular, n is a zero mean Gaussian r.v. with variance $N_oT/4$ and is independent of all other r.v.'s. The term due to the multiple-access noise is

$$u = \sum_{k=2}^{K} w_k\cos\phi_k \tag{5}$$

$$w_k = b_{k,-1}\, R_{k,1}(\tau_k) + b_{k,0}\, \hat{R}_{k,1}(\tau_k) \tag{6}$$

where $R_{k,1}(\tau)$ and $\hat{R}_{k,1}(\tau)$ are the continuous-time partial cross-correlation functions defined by

$$R_{k,1}(\tau) = \int_0^{\tau} a_k(t-\tau)\, a_1(t)dt; \quad \hat{R}_{k,1}(\tau) = \int_{\tau}^{T} a_k(t-\tau)a_1(t)dt \tag{7}$$

In this paper system performance is determined by the error probability $P(\varepsilon)$ for the first user. From here on we assume $b_{1,0} = 1$ and compute

$P(Z_1<0)$. By a symmetry argument, this equals the error probability $P(\varepsilon)$. Hence

$$P(\varepsilon) = P(\sqrt{P/2}\,T + \sqrt{P/2}\,u + n < 0) = \int Q\left(\sqrt{\frac{2PT}{N_o}}\left(1 + \frac{u}{T}\right)\right) f_u(u)\,du. \tag{8}$$

Here $PT/N_o \overset{\Delta}{=} E/N_o$ is called the signal to Gaussian noise ratio and $E \overset{\Delta}{=} PT$ is the energy per bit. As mentioned before, it is not feasible to try to calculate the density of u. It is, however, feasible to compute the moments of u and the two methods described in this paper approximate $P(\varepsilon)$ using only these moments.

CALCULATION OF THE MOMENTS

We wish to find the moments of the multiple-access noise term u defined in eq. (5). As will be shown later, all the odd moments of w_k are zero. Let $u_j = \sum_{k=2}^{j} w_k \cos\phi_k = u_{j-1} + w_j\cos\phi_j$, for $j\geq 3$, be the partial sum of the first j-1 multiple-access interferences. It follows by induction that all the odd moments of u_j are zero. Since w_j and ϕ_j are independent, the even central moments of u_j are

$$E[u_j^{2n}] = \sum_{i=0}^{n} \cdot \binom{2n}{2i} E[u_{j-1}^{2i}]\, E[w_j^{2n-2i}]\, E[\cos^{2n-2i}\phi_j] \tag{9}$$

Hence the even central moments of u can be evaluated recursively from the individual even central moments of $\cos\phi_K$ and w_k which are given below.

$$E[\cos^{2K}\phi_j] = \binom{2k}{k} 2^{-2k} \tag{10}$$

In eq. (6), w_k is a function of the three r.v.'s $b_{k,-1}$, $b_{k,0}$ and τ_k, whose joint p.d.f. is known, therefore it is possible to evaluate its p.d.f. Given the values of the r.v.'s $b_{k,-1}$ and $b_{k,0}$, it can be shown that w_k can be expressed in terms of the periodic cross-correlation function $\theta_{k,1}$, and the odd cross-correlation function $\hat{\theta}_{k,1}$ defined by Massey and Uhran [8]. For $\ell T_c \leq \tau_k < (\ell+1)T_c$

$$w_k = \begin{cases} \theta_{k,1}(\ell)T_c + [\theta_{k,1}(\ell+1)-\theta_{k,1}(\ell)](\tau_k-\ell T_c); & b_{k,-1}=1, b_{k,0}=1 \quad \text{(11a)} \\ \hat{\theta}_{k,1}(\ell)T_c + [\hat{\theta}_{k,1}(\ell+1)-\hat{\theta}_{k,1}(\ell)](\tau_k-\ell T_c); & b_{k,-1}=-1, b_{k,0}=1 \quad \text{(11b)} \\ -\hat{\theta}_{k,1}(\ell)T_c - [\hat{\theta}_{k,1}(\ell+1)-\hat{\theta}_{k,1}(\ell)](\tau_k-\ell T_c); & b_{k,-1}=1,\ b_{k,0}=-1 \quad \text{(11c)} \\ -\theta_{k,1}(\ell)T_c - [\theta_{k,1}(\ell+1)-\theta_{k,1}(\ell)](\tau_k-\ T_c); & b_{k,-1}=-1,\ b_{k,0}=-1 \quad \text{(11d)} \end{cases}$$

where given the two codewords (a_k^j) and (a_l^j),

$$\theta_{k,l}(\ell) = \sum_{j=0}^{N-1} a_k^j \, a_l^{(j+\ell)\mathrm{MOD}\ N}; \qquad 0 \le \ell \le N-1 \tag{12}$$

$$\hat{\theta}_{k,l}(\ell) = C_{k,l}(\ell) - C_{k,l}(\ell-N) \tag{13}$$

and

$$C_{k,l}(\ell) = \begin{cases} \sum_{j=0}^{N-1-\ell} a_k^j \, a_l^{j+\ell}; & 0 \le \ell \le N-1 \\ \sum_{j=0}^{N-1+\ell} a_k^{j-\ell} \, a_l^j; & 1-N \le \ell \le 0 \\ 0 \quad ; & |\ell| > N \end{cases} \tag{14}$$

$C_{k,l}(\ell)$ is called the discrete aperiodic cross-correlation function. It follows from the fact that eq. (11a) and (11b) are the negatives of eq. (11c) and (11d) respectively that $f_{w_k}(w)$ is an even function. Let us note that the moments of u depend only on the set of codewords $\{(a_k^j)\}$.

SERIES EXPANSION METHOD

Following [4,5] we expand the Q function in eq. (8) in a Taylor series about $\sqrt{2PT/N_o}$, take the expection of each term, and obtain a series representation of the average error probability.

$$P(\varepsilon) = Q\left(\sqrt{\frac{2PT}{N_o}}\right) + \frac{e^{\frac{-PT}{N_o}}}{\sqrt{\pi}} \sum_{n=1}^{\infty} \frac{1}{(2n)!} H_{2n-1}\left(\sqrt{\frac{PT}{N_o}}\right)\left(\frac{PT}{N_o}\right)^n \frac{E[u^{2n}]}{T^{2n}} \tag{15}$$

where $H_n(x)$ is the Hermite polynomials which can be computed recursively by $H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x)$, and $H_o(x) = 1$, $H_1(x) = 2x$. The first term of the series is the error probability without multiple-access noise, and the rest of the series is a function of the signal to Gaussian noise ratio and the even central moments of the multiple-access noise.

The series expansion of the average error probability can be shown to be convergent absolutely [5]. An approximation to the average error probability is obtained by truncating this series, and the truncation error can be absolutely bounded by

$$|R_{2m}| = |\sum_{n=m}^{\infty} P_{2n}| \leq \frac{\frac{1}{\sqrt{2\pi}} \exp(-\frac{PT}{2N_o}) \frac{1}{\sqrt{2m-1}\ m!} (\frac{PT}{N_o})^m \frac{E[u^{2m}]}{T^{2m}}}{1 - \frac{1}{m+1} (\frac{PT}{N_o}) \frac{E[u^{2m+2}]}{T^2 E[u^{2m}]}} \tag{16}$$

where

$$P_{2n} = \frac{e^{\frac{-PT}{N_o}}}{\sqrt{\pi}} \frac{1}{(2n)!} H_{2n-1}(\sqrt{\frac{PT}{N_o}}) (\frac{PT}{N_o})^n \frac{E[u^{2n}]}{T^{2n}} \tag{17}$$

If the ratio of multiple-access to Gaussian noise is high (e.g., if there are many users) or if the signal to Gaussian noise ratio is high, the series may converge slowly [5]. In these cases, so many terms may have to be included in the series that it may become computationally infeasible. It may be, however, that the Gauss quadrature rule, which is explained in the next section, can still be applied.

GAUSS QUADRATURE RULE

The Gauss quadrature rule (GQR) is a more efficient procedure for evaluating the average error probability than the series expansion method. Since the average error probability is an integral with the p.d.f. of u as a weight function, it can be approximated using the GQR as a linear combination of a finite number of values of the Q function as

$$P(\varepsilon) = \int Q(\sqrt{\frac{2PT}{N_o}}(1 + \frac{u}{T}))f_u(u)du \cong \sum_i f_i Q(\sqrt{\frac{2PT}{N_o}} (1 + \frac{u_i}{T})) \tag{18}$$

In eq. (18) $\{f_i\}$ are called the weights and $\{u_i\}$ are called the abscissas associated with the weight function $f_u(u)$. The algorithm of Golub and Welsch [7] uses the first 2n+1 moments of u to compute a set of n weights and n abscissas, which can then be used to calculate the average error probability. Let us note that computation of the u_i's and f_i's does not depend on the form of the Q function or on PT/N_o. The same technique is used in [6] to evaluate the error probability in systems with intersymbol interference and additive noise. The mathematical details are shown in [7]. We will summarize the terminology and outline the algorithm.

We begin describing some properties of the weight function. Given a weight function w(x) in the interval [a,b], it is possible to define a sequence of polynomials $\{p_i(x)\}$ that are orthonormal with respect to w(x) and such that the degree of $p_n(x)$ equals n, i.e.,

$$\int_a^b p_n(x)p_j(x)w(x)dx = \delta_{nj} \tag{19}$$

Moreover, a three term recurrence relation holds for this family of orthonormal polynomials

$$xp_{j+1}(x) = \beta_j p_j(x) + \alpha_j p_{j-1}(x) + \beta_{j-1}p_{j-2}(x), \quad j = 1,2,\ldots,n \tag{20}$$

$$p_{-1}(x) = 0,$$

and $p_n(x)$ has n distinct real roots, all of which lie in the interval $[a,b]$.

Theorem. Let $w(x) \geq 0$ be a weight function defined on $[a,b]$ associated with orthonormal polynomials $\{p_n(x)\}$. Let the zeros of $p_n(x)$ be $a < x_1 < x_2 < \ldots < x_n < b$. A set of positive constants w_1, w_2, $\ldots$, w_n, and ξ, $a \leq \xi \leq b$, can be found such that for any 2n-th order continuously differentiable function $f(x)$ on $[a,b]$.

$$\int_a^b f(x)w(x)dx = \sum_{j=1}^{n} w_j f(x_j) + \frac{f^{(2n)}(\xi)}{(2n)!\, k_n^2}; \quad a \leq \xi \leq b \tag{21}$$

where

$$w_j = -\frac{k_{n+1}}{k_n}\frac{1}{p_{n+1}(x_j)p_n'(x_j)}, \quad p_n'(x_j) = \left.\frac{dp_n(x)}{dx}\right|_{x=x_j}; \quad j=1,2,\ldots n;$$

and k_n is the leading coefficient of the polynomial $p_n(x)$.

In fact, the algorithm of Golub and Welsch can be used to compute the set of abscissas and weights from only the moments of u. We outline the two major steps.

(i) evaluate the coefficients of the three term recurrence relation by means of the first 2n+1 moments of the weight function $w(x)$. This step consists of the Cholesky decomposition of a positive definite matrix M formed by the first 2n+1 moments of $w(x)$.

(ii) based on the coefficients of the three term recurrence relation, form a symmetrical tridiagonal matrix, then the abscissas are the corresponding eigenvalues and the weights are the first components of the eigenvectors of this tridiagonal matrix.

The accuracy of the GQR increases with n. Indeed, larger n's are required to maintain accuracy as the multiple-access noise or signal to Gaussian noise ratio increases. Unfortunately, as n increases, the complexity also increases and in addition the Cholesky decomposition becomes numerically unstable. Thus, although the GQR works in many cases where the series expansion method does not, it also has limitations with regard to multiple-access noise and signal to Gaussian noise ratio.

GAUSSIAN ASSUMPTION

We now review a commonly applied approximate method of estimating performance of DS-SSMA systems. When there are sufficient number of users in the system, the p.d.f. of the multiple-access noise u approaches the Gaussian distribution with the variance equal to the sum of the variances of the Gaussian noise and the multiple-access noise. The average error probability is then

$$P(\varepsilon) = Q\left(\frac{\sqrt{P/2}\ T}{(N_0T/4 + P/2\ E[u^2])^{\frac{1}{2}}}\right) \tag{22}$$

where

$$E[u^2] = E[(\sum_{k=2}^{K} w_k\cos\phi_k)^2] = \sum_{k=2}^{K} E[w_k^2]\ E[\cos^2\phi_k]$$

NUMERICAL RESULTS

The series expansion method, the GQR, and the Gaussian assumption were used to evaluate average error probability in first example of DS-SSMA systems with 31 users. First, 31 codewords of length 127 were chosen arbitrarily from the AO/LSE and LSE/AO sequences described in [8] and assigned to the 31 users. Then for each k, 1 ≤ k ≤ 31, the average error probability was calculated as a function of E/N_0 for user number 1 when users 1,2,...,k are transmitting simultaneously. In Fig. 1 we have plotted the results of these calculations using the GQR with 12 terms and 25 moments. The accuracy is believed to be to two significant figures. The average error probability was also calculated using the series expansion method with 23 terms and 45 moments. For small values of E/N_0, these results agreed with the results of the GQR to two significant figures. However, as E/N_0 increased there came a point where they only agreed to one significant figure, and for slightly larger E/N_0 they disagreed sharply. To indicate where the series expansion method loses accuracy we have drawn a dashed line in Fig. 1 corresponding to where the series expansion method differs from the GQR by roughly 10%. To the left of this dashed line the series expansion method is quite accurate, while to the right its accuracy deteriorates rapidly. We have also plotted the results based on the Gaussian assumption in Fig. 1. It can be seen that the Gaussian assumption gives fairly good results, especially when the number of users is large and E/N_0 is small.

The same set of calculations were performed for a set of 31 Gold sequences with length 127 and are plotted in Fig. 2.

We note that with both the series expansion method and the GQR, the points of the curves in Figures 1 and 2 did not have to be computed independently. For example, the moments of u depend only on the codewords. Once they are calculated, an entire curve of P(ε) versus E/N_o may be calculated, Furthermore, the moments for k+1 users can be computed recursively from the moments for k users.

In comparison to previous methods both the GQR and the series expansion method allow the analysis of the average error probability to be carried out for a larger number of users and for larger E/N_o. In particular, the GQR has a potential to be a useful tool in the evaluation of average error probability in large DS-SSMA systems.

References

[1] M. B. Pursley, "Performance evaluation for phase coded SSMA communication-Part I: System Analysis" IEEE Trans. COM-25, No. 8, Aug. 1977, pp. 795-799.

[2] K. Yao, "Error probability of asynchronous SSMA communication systems" IEEE Trans. COM-25, No. 8, Aug. 1977, pp. 803-809.

[3] D. E. Borth, M. B. Pursley, D. V. Sarwate, W. E. Stark, "Bounds on error probability for DS-SSMA communications", MIDCON conference, paper 15/1, Nov. 1979.

[4] O. Shimbo, M. I. Celebiler, "The probability of error due to intersymbol interference and Gaussian noise in digital communication systems", IEEE Trans. COM-19, April 1971, pp. 113-119.

[5] E. Y. Ho, Y. S. Yeh, "A new approach for evaluating the error probability in the presence of intersymbol interference and additive Gaussian noise", B.S.T.J. Vol. 49, No. 9, Nov. 1970, pp. 2249-2266.

[6] S. Benedetto, G. DeVincetiis, A. Luvinson, "Error probability in the presence of intersymbol interference and additive Gaussian noise for multilevel digital signal", IEEE Trans. COM-21, No. 3, March 1973, pp. 181-190.

[7] G. H. Golub, J. H. Welsch, "Calculation of Gauss Quadrature rule", Math. Compt., Vol. 23, April 1969, pp. 221-230.

[8] J. L. Massey, J. J. Uhran, "Sub-baud coding", Proceedings of the Thirteenth Annual Allerton Conference on Circuit and System Theory, Oct. 1975, pp. 539-547.

[9] M. B. Pursley, H. F. A. Roefs, "Numerical evaluation of correlation parameters for optimal phases of binary shift-register sequence", IEEE Trans. COM-27, No. 10, Oct. 1979, pp. 1597-1604.

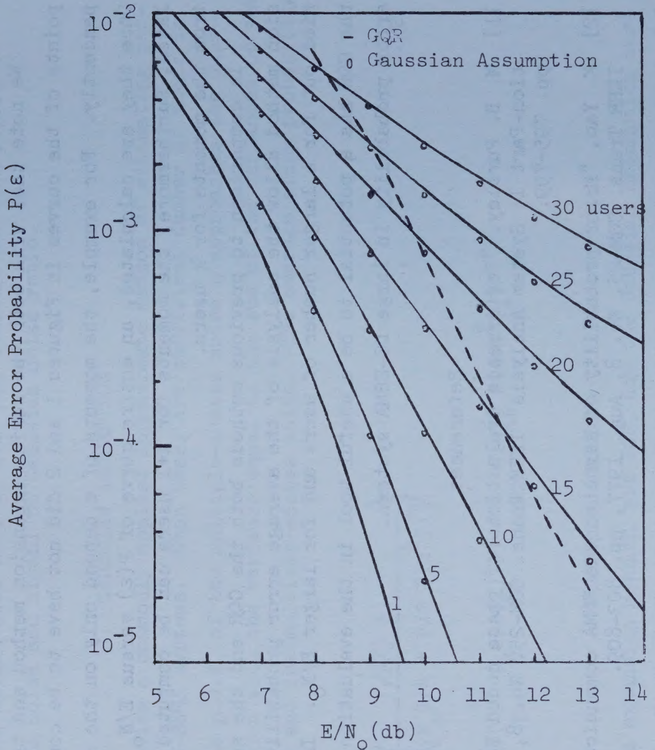


Figure 1. $P(\varepsilon)$ vs. E/N_o for various Number of Users

(Dashed line shows where the series expansion method differs by 10% from GQR)

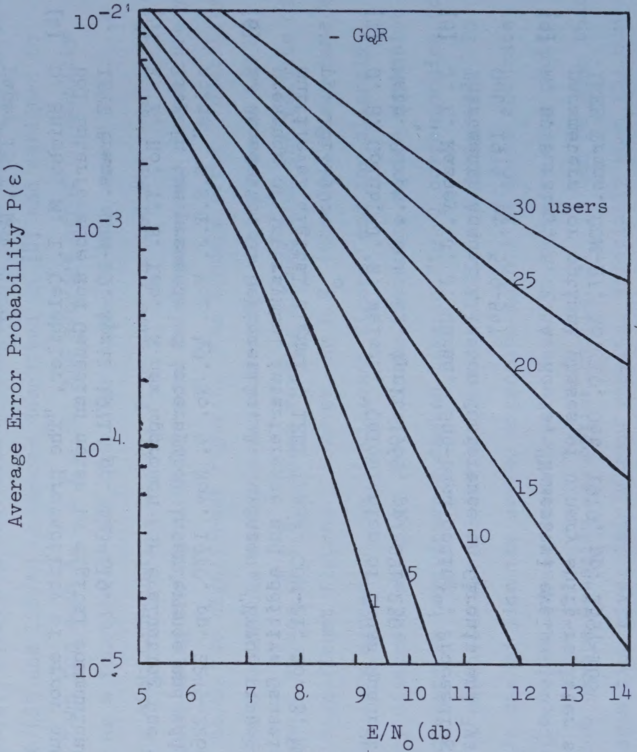


Figure 2. $P(\varepsilon)$ vs. E/N_o for various Number of users

PROGRESS IN DIGITAL MODULATION TECHNIQUES FOR SATELLITE SYSTEMS

W.W.WU. and T. R. ROWBOTHAM
International Telecommunication Satellite Organization
Washington D.C. 20024

ABSTRACT

This paper presents three recent results on digital modem development under an realistic nonlinear satellite channel environment. A 120 Mbps QPSK modem, operated with burst mode, achieves an equivalent power loss of 1.70 dB at 10^{-4} bit error rate. An modified maximum likelihood sequence detector is shown to exhibit about 1.0 dB improvement at the same BER. A 3.5 dB E_b/N_o gain over conventional QPSK in simulation is obtainable by a coded phase modem operated with co-channel and adjacent channel interferences.

1.0 INTRODUCTION

INTELSAT has been sponsoring modem development for satellite communications since 1965. From both power-bandwidth efficiency and performance viewpoints QPSK modems have been predominate. In recent years the emphasis has shifted to high bit rates for time division multiple access systems. In addition futher performance improvements are sought as channel impairments become understood, or operating requirement changes.

The first really significant work involved field tests of a 4 - phase PSK -TDMA modem using the Paumalu earth station and INTELSAT IV F-4. 68 Mbit/s transmission rate was used in an investigation of the various earth station and transponder transmission path impairments. Further work at about the same time (1974) was conducted in exploration of the possibility of using 8-phase and 16-phase PSK-TDMA. An experimental model 1 Gbit/s modem was also constructed and evaluated. However, almost all the modem designs were based on a linear satellite channel and significant performance degradations have been experienced when the modems were subjected to non-linear channel conditions.

From a modem point of view, the model of a simplied satellite channel is shown in Figure 1. The satellite consists of input, output filters and a traveling wave tube (TWT). The transmitting earth station has a modulator, a transmitting filter and a high power amplifier (HPA). The receiving station contains a receiving filter, an aperture equalizer and the corresponding demodulator. The nonlinear characteristics of HPA and TWT (both amplitude and phase) are shown in Figure 2.

With 0 dB denotes at saturation and other dB values refered as backoffs from saturation point, the notation (HPA/TWT) indicates the corresponding operating point when they are in concatenation. For power efficiency it is desirable to operate both devices in saturation. In section 2.0 a 120 Mbps QPSK modem, operated with burst mode, will be described and hardware test results will be presented. The simulation results of a modified maximum likelihood sequence detector will be discussed in section 3.0. Simulation results will be reported in section 4.0 for a coded phase modem which can provide significant performance improvement without the compromising of transmission efficiency.

2.0 A 120 Mbps NONLINEAR CHANNEL MODEM

In a linear channel the receiver matched filter acts on the signal to minimize intersymbol interference (ISI). In a non-linear environment, the original signal spectrum is almost completely regenerated after each non-linear operation. Hence the matched filter at the receiver fails to produce smooth phase and amplitude variations in the demodulator, and so the condition of minimum intersymbol interference is invalidated. Chakraborty and Noguchi [1] have computed the normalized power spectrum for QPSK using the channel simulation model of Robinson et al [2], after transmission through a typical band limiting network, namely a cosine roll-off Nyquist filter with 45% roll-off, and a non-linear device operating backed-off from saturation.

Although it is desirable to choose a combination of transmit and receive filters which minimize the bit-error rate for a given carrier-to-noise ratio, under given conditions of nonlinearity in the HPA and TWT, it is also necessary to ensure that the spectral regrowth does not comprise significant adjacent channel interference. Because, in general, the earth station does not provide filtering at the HPA output, and an out-of-band emission limit established. To ensure insignificant adjacent channel noise, the out-of-band emission has been determined by INTELSAT as having to be below 4 dBW/ 4 kHz in order to maintain signal quality.

The problem of the uncertainly of the operating points for the HPA and TWTA was overcome to some extent by developing a 'nonlinear' modem which would yield near-optimum performance over a range of HPA and TWTA backoffs. This modem, which was developed for INTELSAT by Nippon Electric Company, Japan, under a contract placed in 1978, had filter characteristics based on extensive computer simulation. Three types of filter combination were used: [3]

(i) Nyquist shaping with aperture equalizer for transmission and wideband noise cut -off filter for receiving;

(ii) Wide band sharp cut-off filter with aperture equalizer and Nyguist shaping filter;

(iii) Wide band sharp cut-off for transmission and Nyquist shaping with an aperture equalizer for receiving.

Over the range (HPA/TWT) back-0ffs from (OdB/OdB) to (14dB/14dB) the values of the cut-off frequency normalized by the Nyquist frequency (f_e) and the roll-off factor (f) were varied to optimize the performance. On this basis, the filters of type (ii) were chosen for implementation. The optimum roll-off, taking into account the adjacent channel interference considerations, is about 45%. The transmit filter was a pass-band flat elliptic filter with delay characteristic equalized, and pass-band amplitude ripple of 0.5dB. The receive filter was implemented as 40% roll-off to err on the safe side if in case the adjacent channel power exceeded the anticipated allowance; for the transmit roll-off 15% was selected, as the smallest that can be implemented in hardware. The results of tests on the modem gave the results shown in Figure 3 for the set of back-off conditions indicated. Figure 3 shows that at, for example at bit-error rate of 10^{-5}, the required E_b/N_o in the modem is less than that of the 'linear' modem by 4.3 dB at (0/0) condition. The output power spectrum of the QPSK modem for the two operating conditions is shown in Figure 4.

For comparison a offset QPSK modem was also constructed by NEC. The performance is shown in Figure 5 with and without added filters. In compare with QPSK the offset QPSK degrades at higher E_b/N_o conditions for the saturation case. With range of backoff conditions from 0 to 14 dB the equivalent power losses of both modem are compared in Table 1 for BER at 10^{-4} and 10^{-6}. These results were obtained with optimized filter characteristics.

Tabe 1 Equivalent Power Loss vs. Input Backoff

Input Backoff (HPA/TWT) dB	QPSK		O-QPSK	
	10^{-4}	10^{-6}	10^{-4}	10^{-6}
14/14	1.10	1.10	1.10	1.10
14/4	1.38	1.58	1.53	1.83
6/4	1.48	1.83	1.88	2.18
6/4	1.55	1.90	2.00	2.35
0/0	1.70	1.10	2.45	2.85

3.0 MAXIMUM LIKELIHOOD SEQUENCE DETECTION (MLSD)

It has been known in theory that MLSD can improve signal quality subject to the intersymbol interference (ISI). Partial response signaling schemes have been suggested to minimize the effect of ISI. The common characteristic of partial response signaling schemes is the utilization of overlapping signal elements, through the controlled ISI thereby increasing the ratio of the transmission symbol rate to the system bandwidth. This is desirable in satellite communication systems which are, in general, bandwidth limited.

Forney [4] not only applied the Viterbi algorithm to intersymbol interference problems, but generated an impact in the field of investigations of the understanding and solutions to such problems. His main contribution was the observation that the Viterbi algorithm can be optimally applied to replace the earlier nonlinear processing techniques required in a partial response receiver. With variations in the same area the other significant contributors are: Magee and Prokis [5] for suggesting an adaption for MLSD, Qureshi and Newhall [6] for incorporating a linear equalizer, Mesiya et al [7] for the evaluation of MLSD in a nonlinear channel, and Herrmann [8] for extensive computer evaluations of Forney's approach in a nonlinear satellite channel.

In March 1979 INTELSAT placed a contract with NEC to evaluate the feasibility of applying a MLSD high speed processor to a nonlinear satellite channel, taking into account the practical constraints with a sequences of three and five symbol memories and both HPA and TWTA operated in saturation, the performances in BER are shown in Fig. 6 under INTELSAT V channel conditions. Discrete MLSD refers the usage of sampled whitened matched filters while continuous version refers to unsampled filtering. As shown, the difference is about 0.5 dB in E_b/N_o. For comparison the nonlinear channel performance is also plotted in the most right hand side.

Unfortunately despite the theoretical attractiveness of the performance improvements for both continuous and discrete MLSD, NEC concluded that, at least at present, high speed burst mode implementation of MLSD hardware is impractical.

It has been estimated by NEC that it requires approximately 32,000 pieces of integrated circuits with 32 kilowatts of power to operate a 5 symbol discrete MLSD with less than 7.0 M baud. With 3 symbol discrete MLSD, although with reduced performance improvement, the complexity reduces to 2,000 IC's with estimated 830 watts of power consumption. The operating speed can be increased to 15 M baud. Most hardware applications of interest to INTELSAT are therefore still beyond the reach of practicality.

3.1 Adaptive Threshold Detection with Estimated Sequence (ATDES) [9]

In order to avoid the practical problems of MLSD, NEC have proposed an simplified alternative called adaptive threshold detector with estimated sequence (ATDES), which determines the optimum threshold of signal detection based upon the tentatively estimated sequence. The principle of ATDES for QPSK works as follows.

Assuming each detected signal is interfered with k-future and k past symbols that is, the channel under consideration has a memory of 2k + 1 symbols by selecting the threshold boundaries on the phase coordinates the intersymbol interference may be eliminated. The effectiveness of the elimination based on the distance between a pair of received signals. The objective of ATDES is to select the optimum threshold boundary using the tentatively estimated sequence. The configuration of an ATDES resembles that of an adaptive equalizer. However ATDES adapts the decision levels.

The ATDES implementation being pursued by NEC considers the intersymbol interference from three past and three future symbols. The past symbols are given from the output of the final decision circuit, and the future is estimated by the tentative decision circuit.

The averaged distortion is stored in the distortion memory circuit and is subtracted from the delayed received signal. In this way, the decision scheme with adaptive threshold control is obtained. The content of the distortion memory circuit is updated by averaging the received intersymbol interferences corresponding to each sequence pattern. The error propagation caused by this tentative decision is considered to be a disadvantage as of ATDES. This problem can be reduced by null zone detection.

Depending on the transmitting and receiving filter characteristics, the performance of ATDES, as well as MLSD vary. An example of 3 and 5 symbol ATDES shown in Fig. 6 which was obtained with Nyquist shaping transmitting filter of 30% roll-off and 10% roll-off receiving filter at 1.2 x Nyguist frequency cut-off. The transmitting filter was equalized At 10^{-4} BER the 5 symbol ATDES performed about 0.5 dB worse than the 5 symbol discrete MLSD in E_b/N_o. Both versions of the ATDES processors appear practical feasible. The 5 symbol ATDES hardware is estimated about 440 IC's with 130 Watts power consumption. The 3 symbol ATDES requires approximately 300 IC's with 100 Watts power. Both can only be operated below 30 Mbaud.

4.0 CODED PHASE MODULATION (CPM)

CPM refers to the technique of combining of channel coding and phase modulation for the purpose of improving transmission link performance. When the received bit signal level is quantized into 2 subdivisions and then set a threshold level for decision, the detection process is referred as soft decision. The idea of soft decision decoding began with Elia's erasure channel. Recent development of soft decision decoding algorithms has been given by Chase, Massey, Weldon, Hartmann and Rudolph.

In principle the gain of soft decision over hard decision (two level) is about 3.0 dB in E_b/N_o. In practice a 2.0 dB or less is achievable. With little or no bandwidth expansion, Anderson and Taylor suggested a class of multi-h phase codes primarily for MFSK. Aulin, Rydbeck, and Sandberg investigated constant envelope, continuous phase modulation schemes.

Underboeck [10] suggested a general mapping technique, which is referred as set partitioning. When this mapping technique combines with the Viterbi algoritm it produces impressive results. The mapper maps the bit sequence from the encoder to multilevel channel signals, which in this case are the number of phases in the modulation under consideration. The mapping approach assumes a suitable trellis state-transition of m bits per symbol. Mapping takes place from each state 2^m possible transitions to the successive states. After the trellis diagram has been selected the assignments of channel signals to the state transitions are required in order to maximize the free Euclidean distance criterion. This maximization of distance through set partition was shown as an example in [10] for 8 phase modulation. The de-mapper used the Viterbi algorithm with soft decision decoding.

Under joint INTELSAT and COMSAT sponsorship, studies, analyses, and computer simulation programs have been developed [11] at COMSAT Labs to apply the set partitioning mapping technique to the INTELSAT channel condition. A rate 2/3 convolutional code was selected for evaluation. Figure 7 shows BER performance with (w) or without (w/o) co-channel interference (CCI), adjacent channel interference (ACI), earth station equalization (Es/EQ) Ideal CPM and ideal QPSK for a channel with only added white Gaussian noise (AWGN). Note that at 10^{-4} bit error rate under the same conditions of co-channel interference and adjacent channel interference, with or without ideal earth station equalization the 8 phase CPM requires about 3.5 dB less E_b/N_o power than QPSK. Particular of interest is the fact that this gain was obtained without the sacrifice of transmission efficiency i.e. the coding rate is unity.

5.0 A CONCLUDING REMARK

This paper reports only the state-of-art on modem developments in INTELSATs environment. Other significant results such as high packing density (4 bits/Hz), high speed (760 Mbps MSK), and spectrum spreading modems have not been addressed.

REFERENCES

1. Chakraborty D., and Noguchi T., Effects of band limiting on Non-linear Digital Satellite Links" I.C.C. 80 Paper 62.3.1

2. Robinson G., Shimbo O., and Fang, R., "PSK Signal power spectrum power spread produced by memoryless non-linear TWT's COMSAT Technical Review Vol. 3, No 2, Fall 1973 pp. 227-256.

3. Final Report on Multi-Mode Modem For Nonlinear Satellite Channel NEC, Japan, 1980.

4. Forney G. D. Jr., "Maximum-Likelihood Sequence Estimation of Digital Sequence in the Presence of Intersymbol Interference, "IEEE Tran. On Information Theory, May 1972.

5. Magee F. R. Jr., and Proakis J. G., "An Estimation of the Upper Bound on Error Probability for Maximum Likelihood Sequence Estimation on Channels having a Finite Duration Pulse Response, "IEEE Trans. on Information Theory, IT-19, September 1973.

6. Qureshi S. U. H. and Newhall E. E., "An Adaptive Receiver for Data Transmission over Time-Dispersive Channels, "IEEE Trans. on Information Theory, July 1973.

7. Mesiya M. F., Mclane P. J., and Campbell L. C., "Maximum Likelihood Sequence Estimation over Bandlimited Nonlinear Channel, "IEEE Trans. on Communications, July 1977.

8. Herrmann G. F., "Performance of Maximum Likelihood Receiver in the Nonlinear Satellite Channel" IEEE Trans. on Communications, Vol. COM-26, No. 3, March 1978.

9. Final Report on Study Phase, Maximum Likelihood Sequence Detection Process/Modem, December, 1979. 2nd Phase Progress Report September 1980.

10 Ungerboeck, G., "Channel Coding with Multilevel/Phase Signals", IBM Zurich Research Laboratory Report RZ 841 (#28387), 1977

11 Fang R., "Private Communications"

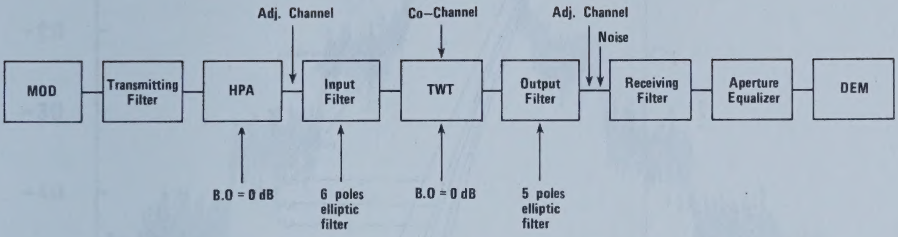

Adj. Channel
Co–Channel
Adj. Channel
Noise
MOD
Transmitting Filter
HPA
Input Filter
TWT
Output Filter
Receiving Filter
Aperture Equalizer
DEM
B.O = 0 dB
6 poles elliptic filter
B.O = 0 dB
5 poles elliptic filter

Figure 1: SATELLITE CHANNEL MODEL

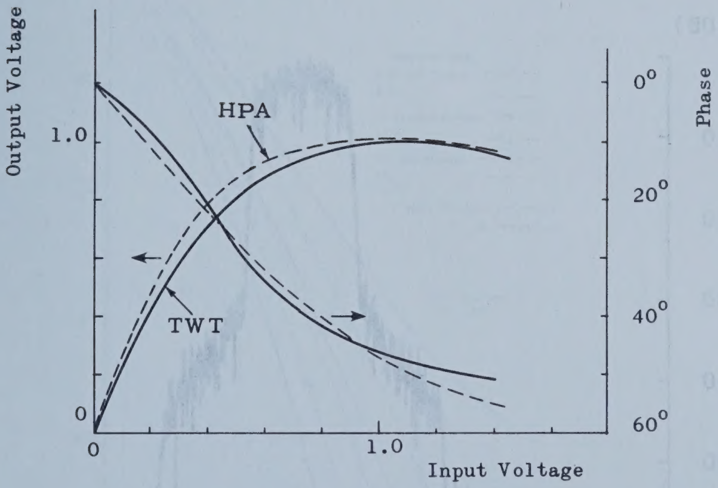


Figure 2: NONLINEAR CHARACTERISTICS OF HPA AND TWT

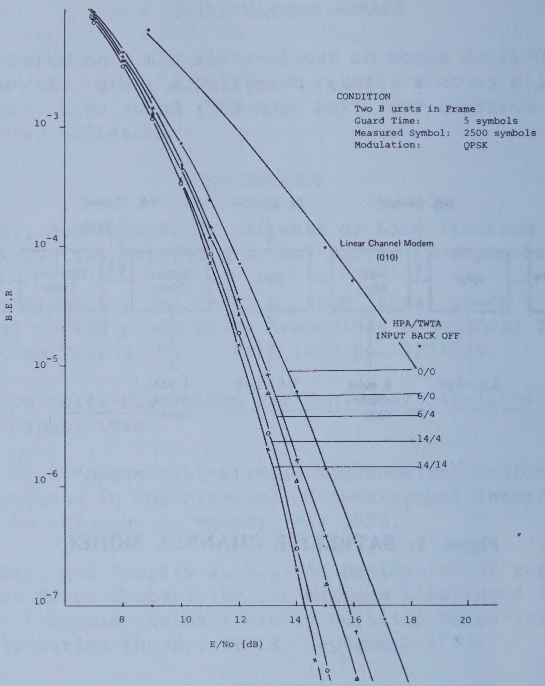


Figure 3: TEST RESULTS IN NON-LINEAR CHANNEL

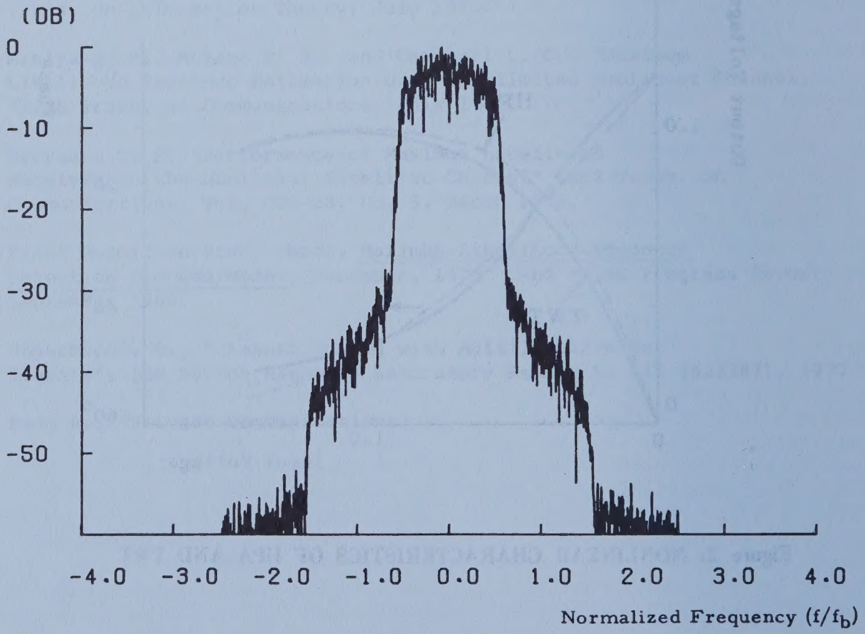


Figure 4: HPA OUTPUT POWER SPECTRUM OF QPSK INPUT BACK-OFF 14 dB, f_b: BAND RATE

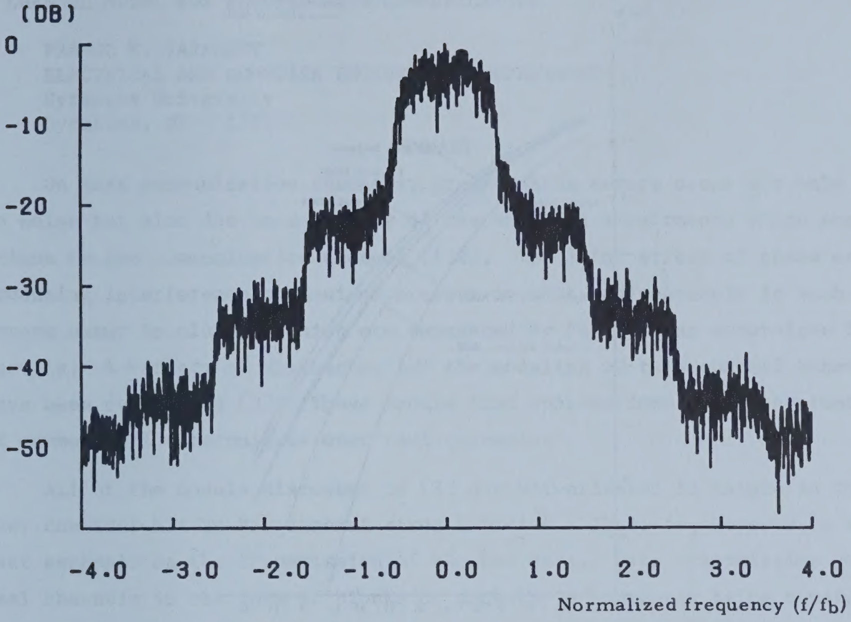


Figure 4: HPA OUTPUT POWER SPECTRUM OF QPSK INPUT BACK-OFF 14 dB, f_b: BAND RATE

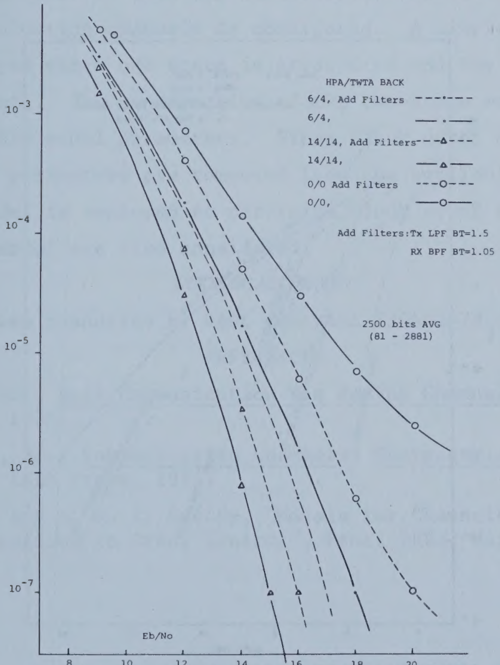


Figure 5: BER vs EB/NO TEST IN BURST MODE OFFSET QPSK MODE

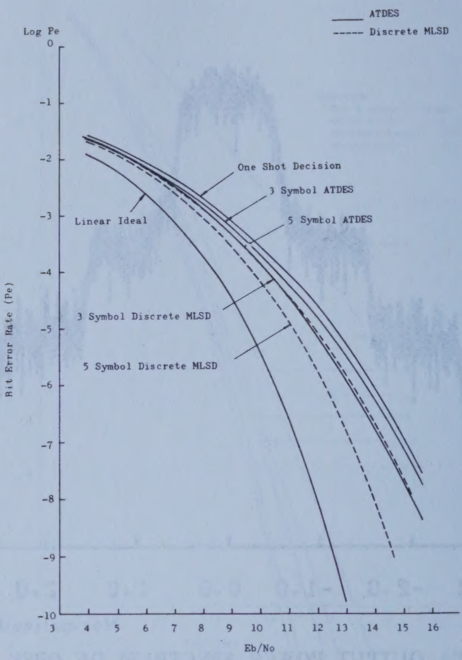


Figure 6: DISCRETE MLSD AND ATDES PERFORMANCE

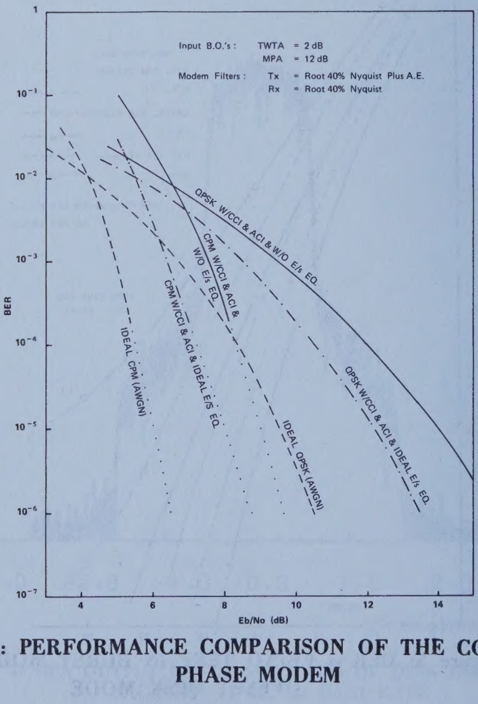


Figure 7: PERFORMANCE COMPARISON OF THE CODED PHASE MODEM

A CHANNEL MODEL FOR BLOCKED DATA COMMUNICATION

PRAMOD K. VARSHNEY
ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
Syracuse University
Syracuse, NY 13210

SUMMARY

On most communication channels, transmission errors occur not only due to noise but also due to a variety of transmission impairments which are unique to the communication channel [1,2]. The joint effect of these error-producing interference mechanisms present on most real channels is such that errors occur in clusters which are separated by fairly long error-free intervals. A variety of approaches for the modeling of this channel behavior have been considered [3]. These models find applications in the evaluation of communication techniques over real channels.

All of the models discussed in [3] are bit-oriented in nature in that they consider bit-by-bit channel error behavior. There is, however, a recent emphasis on the transmission of blocked data. Data transmission over real channels in the form of blocks or packets is currently being considered, e.g. packet radio networks. In this paper, we suggest a block-oriented channel model in that the characterization of the block error behavior of communication channels is considered. A simple two-state Markov model is discussed where one state is error-free and the other corresponds to the error state. The construction of the two-state model requires the computation of the model parameters. Since block error data is not available, the model parameters are computed from the available bit error statistics. The model is employed to determine block error statistics. Applications of the model are also considered.

ACKNOWLEDGEMENT

This work was supported by RADC contract F30602-78-C-0083.

REFERENCES

1. K. Brayer, Ed., Data Communication Via Fading Channels, New York: IEEE Press, 1975.
2. B. Goldberg, Ed., Communication Channels: Characterization and Behavior, New York: IEEE Press, 1976.
3. L. N. Kanal and A. R. K. Sastry, "Models for Channels with Memory and Their Applications to Error Control", Proc. IEEE, Vol. 66, pp. 724-744, July 1978.

UNSTABLE SYSTEMS AND FEEDBACK:THE GAP METRIC

G. ZAMES and A. EL-SAKKARY
Department of Electrical Engineering
McGill University
Montreal, Que., Canada

ABSTRACT

This paper is concerned with unstable input-output systems which are stabilized by feedback in the presence of uncertainty. A metric on unstable systems in terms of the gap between their graphs in a Hilbert Space is introduced. It is shown that the tolerable open-loop uncertainties are precisely those that are small in gap, and that every "continuously robust" metric on unstable systems is equivalent to the gap metric.

1. INTRODUCTION

(This paper is based on El-Sakkary [1]. The main results are stated without proof. Proofs can be obtained from the authors.)

In this paper we consider a system represented by an input-output mapping $\underline{K}$, which is subject to uncertainty introduced during the identification or realization of $\underline{K}$, and which is stabilized by feedback around $\underline{K}$. We seek answers to the questions:What types of uncertainity in $\underline{K}$ can be tolerated without destroying closed-loop stability? In which sense should the uncertainity in $\underline{K}$ be "small" in order to achieve a "small" uncertainity in the closed loop?

We seek an input-output description of the tolerable uncertainities. To this end, we shall introduce a metric on unstable systems in terms of the gap (or aperture) between their graphs, and show that the tolerable small uncertainties are those, and only those, that are small in gap.

The motivation for this work comes from examples involving single-input single-output systems represented by frequency responses belonging to, say, the space $\hat{\mathbb{R}}_o$ of rational, strictly proper functions of the complex variable $s \overset{\Delta}{=} \sigma + j\omega$, which have no $j\omega$-axis poles. Consider the unstable frequency response $\hat{K}_1 \in \hat{\mathbb{R}}_o$, $\hat{K}_1(s) = (s - \frac{1}{2})^{-1}$ which is stabilized by feedback to give $[1 + \hat{K}_1(s)]^{-1} = (s + \frac{1}{2})^{-1}$. If $\hat{K}_1$ is approximated by the response $\hat{K}$ where $\hat{K}(s) = \hat{K}_1(s) + \in (s-1)^{-1}$, the error or uncertainty $\in (s-1)^{-1}$ creates a pole-zero pair close to the point $s=1$, and makes $[1 + \hat{K}_1(s)]^{-1}$ unstable for all small $\in > 0$. Such an uncertainty changes the closed-loop behavior from stable when $\in = 0$ to unstable when $\in > 0$, no matter how small $\in$ is, and clearly can not be tolerated in practical situations. On the other hand, small errors in pole position, ex., where $\hat{K}(s)$ has the form $(s-\frac{1}{2}+\epsilon)^{-1}$, merely create small errors in the closed-loop frequency response without affecting stability, and can be tolerated provided that ϵ is small enough. Our objective is to characterize the tolerable errors and separate them from the others.

In the preceeding example it can be shown (using simple Nyquist theory [2]) that no uncertainty at all can be tolerated as to the number of $\mathrm{Re}(s) > 0$ poles of $\hat{K}(s)$ for feedback control to be feasible.

The situation is different for frequency responses that are stable in the open-loop. For any desired accuracy in the closed-loop in the presence of, say, square-integrable inputs, it is enough to get a sufficiently close approximation to the open-loop frequency response $\hat{K}_1(s)$, and the pole-zero configuration of $\hat{K}_1(s)$ is immaterial. This represents a special case of the following well known continuity principle (in a more general setting): If $\mathcal{X}$ is a normed linear space, and K and $(I + K)^{-1}$ are bounded operators taking $\mathcal{X}$ into $\mathcal{X}$, then the mapping that takes (open-loop) operators K into (closed-loop) operators $(I + K)^{-1}$ is continuous at K in the operator norm induced by $\mathcal{X}$; i.e., given any tolerance $\varepsilon > 0$ on the closed-loop there is a tolerance $\delta > 0$ on the open-loop with the property that if $\|K-K_1\| < \varepsilon$ then $\|[I + K]^{-1} - [I + K_1]\| < \varepsilon$.

However, if K is an operator representing an unstable system, then K is not defined on all of $\mathcal{X}$, but on some proper subspace of $\mathcal{X}$ consisting of those inputs that produce (finite-norm) outputs in $\mathcal{X}$. Here, the operator norm is not defined for K, and the continuity principle as stated above is not applicable. In that case, it would be desirable to have some alternative to the norm for measuring approximations or uncertainty, which would allow a modified version of the continuity principle to be stated.

We shall show that the gap metric, to be introduced here, is suitable for this purpose and, moreover, gives an input-output characterization of the tolerable uncertainties, precisely separating them from the others.

2. CLOSED OPERATORS BOUNDED ON THEIR DOMAINS:DEFINITIONS

$\mathcal{H}$ is a Hilbert space. $\mathcal{H}_e$ is an extended space obtained from $\mathcal{H}$ via a family of projection operators $P_t:\mathcal{H} \to \mathcal{H}$ depending on the parameter $t \geqslant 0$.

$\underline{\mathbb{A}}$ is the algebra with identity $\underline{I}$ of causal linear mappings $\underline{K}:\mathcal{H}_e \to \mathcal{H}_e$ representing input-output systems. The elements of $\underline{\mathbb{A}}$ will be denoted by underlined capitals and referred to as *systems*. $\mathbb{B}$ is the normed subalgebra of $\underline{\mathbb{A}}$ consisting of *stable* systems, i.e., those $\underline{H} \in \underline{\mathbb{A}}$ for which the supremum

$\sup \{\|Hx\|/\|x\| : x \in \mathcal{H},\ x \neq 0\}$ is finite. The norm of $\underline{H} \in \underline{\mathbb{B}}$ is given by that supremum.

The *stable subspace*, denoted by $St(\underline{K})$, of any $\underline{K} \in \underline{\mathbb{A}}$ is the subspace of $\mathcal{H}$ defined by the equation $St(\underline{K}) = \{x \in \mathcal{H}: \underline{K}\, x \in \mathcal{H}\}$. If $St(\underline{K})$ is a proper subspace of $\mathcal{H}$ then some inputs in $\mathcal{H}$ are mapped into outputs in $\mathcal{H}_e$ but not in $\mathcal{H}$, and $\underline{K}$ is not stable.

We shall be interested in feedback systems represented by mappings in $\underline{\mathbb{A}}$ of the form $(\underline{I} + \underline{K})^{-1}$, which are stable, but in which $\underline{K}$ is not necessarily stable. From the stability and causality of $(\underline{I} + \underline{K})^{-1}$ it can be shown that the complete mapping $(\underline{I} + \underline{K})^{-1}$ is determined by its restriction to $\mathcal{H} \times \mathcal{H}$, i.e., by those input-output pairs $(u, (\underline{I} + \underline{K})^{-1}u)$ for which both u and $(\underline{I}+\underline{K})^{-1}u$ belong to $\mathcal{H}$. Such a restriction is a function from $\mathcal{H}$ into $\mathcal{H}$ whose domain is $St(\underline{K})$. Functions of this kind will be called operators. More specifically, an *operator* $K: \mathcal{H} \to \mathcal{H}$ is a function whose domain is a subspace of $\mathcal{H}$, and which is the restriction of some system $\underline{K} \in \underline{\mathbb{A}}$ to $\mathcal{H} \times \mathcal{H}$. We shall use the same capital letter to denote a system and its associated operator, but the former will be underlined whereas the latter will not. The stable subspace $St(K)$ of K is defined to be equal to $St(\underline{K})$.

Let $\mathbb{A}$ be the set of all operators that are restrictions of systems in $\underline{\mathbb{A}}$. If K_1 and K_2 are operators in $\mathbb{A}$, their sum $(K_1 + K_2)$ in $\mathbb{A}$ is defined to be the restriction of the sum $(\underline{K}_1 + \underline{K}_2) \in \underline{\mathbb{A}}$ to $\mathcal{H} \times \mathcal{H}$. Similarly, the other algebraic operations in $\mathbb{A}$ are defined to be restrictions to $\mathcal{H} \times \mathcal{H}$ of systems obtained by the corresponding operations in $\underline{\mathbb{A}}$. $\mathbb{A}$ is now an *algebra*, consisting of operators with restricted domains. However, it should be remembered that not all such operators belong to $\mathbb{A}$, but only those that are restrictions of systems in $\underline{\mathbb{A}}$. (Without such a qualification, the set of operators would not be an algebra).

Let the product space $\mathcal{H} \times \mathcal{H}$ be made into a Hilbert-space, with the inner product on $\mathcal{H} \times \mathcal{H}$ derived from $\mathcal{H}$. If $K \in \mathbb{A}$ is any operator, the term *graph* of K, denoted by $\Gamma(K)$, will denote the set of points (u, Ku) *viewed as a subspace of the Hilbert space* $\mathcal{H} \times \mathcal{H}$.

Prop 1 If $(\underline{I} + \underline{K})^{-1} \in \underline{\mathbb{A}}$ is *stable* then $\Gamma(K)$ is a closed subspace of $\mathcal{H} \times \mathcal{H}$.

An operator K is called *closed* if its graph is a closed subspace of $\mathcal{H} \times \mathcal{H}$. It follows from Prop 1 that the only operators that can generate stable feedback systems, and therefore the only ones we need to consider in our problem, are the closed operators.

An operator $K \in \underline{\mathbb{A}}$ will be called *domain-bounded* if there is a constant $c \geqslant 0$ with the property that $\|Ku\| \leqslant c\|u\|$ for all u in $St(K)$.

Let $\mathbb{B}_D$ *denote the subset of closed and domain-bounded operators in* $\mathbb{A}$, and $\mathbb{B}_D$ the corresponding subset of systems in $\underline{\mathbb{A}}$. In this paper we shall confine our attention to systems in $\underline{\mathbb{B}}_D$.

Frequency-response systems

Let $\mathbb{R}_o$ denote the algebra of systems $\underline{K}:L_{2e}(0,\infty) \to L_{2e}(0,\infty)$, $\widehat{\underline{K}u}(s) = \hat{K}(s)\hat{u}(s)$, in which $\hat{}$ denotes the Laplace transformation, and $\hat{K}(s)$ is a rational, strictly causal frequency response in $\hat{\mathbb{R}}_o$ without $j\omega$-axis poles. Let $\mathbb{R}_o$ be the corresponding algebra of operators.

Prop 2 The operators in $\mathbb{R}_o$ are closed and domain-bounded, i.e., belong to $\mathbb{B}_D$.

3. THE GAP METRIC

The gap (or aperture) is an angular measure of the distance between closed subspaces of a Hibert space $\mathcal{H}$, and is defined as follows.

The distance $d(x,\mathcal{H})$ between an element $x \in \mathcal{H}$ and a subset $\mathcal{S} \subset \mathcal{H}$ is $d(x,\mathcal{H}) \triangleq \inf_{y\in\mathcal{S}} \|x-y\|$. The distance between any two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ of $\mathcal{H}$ is $d(\mathcal{S}_1,\mathcal{S}_2) \triangleq \sup_{x\in\mathcal{S}_1} d(x,\mathcal{S}_2)$.

If $\mathcal{M}$ and $\mathcal{N}$ are closed subspaces of $\mathcal{H}$, the directed gap from $\mathcal{M}$ to $\mathcal{N}$ is

$$\vec{\delta}(\mathcal{M},\mathcal{N}) \triangleq \sup_{\substack{x\in\mathcal{M}\\ \|x\|=1}} d(x,\mathcal{N})$$

i.e., $\vec{\delta}(\mathcal{M},\mathcal{N})$ is the distance from the unit sphere in $\mathcal{M}$ to the subspace $\mathcal{N}$. The gap between $\mathcal{M}$ and $\mathcal{N}$ is $\delta(\mathcal{M},\mathcal{N}) \triangleq \max\,[\vec{\delta}(\mathcal{M},\mathcal{N}),\ \vec{\delta}(\mathcal{N},\mathcal{M})]$. For lines in the Euclidean plane, the gap is simply the arcsine of the acute angle between them.

The gap between two systems in $\underline{\mathbb{B}}_D$, or between their operators in $\underline{\mathbb{B}}_D$, is defined to be the gap between their graphs, viewed as closed subspaces of the product Hilbert space $\mathcal{H}\times\mathcal{H}$.

Prop 3 The gap is a metric for $\underline{\mathbb{B}}_D$ (or $\mathbb{B}_D$). For (stable) systems in $\underline{\mathbb{B}}$ there are constants $\varepsilon > 0$ and $c(\varepsilon) \geqslant 0$ for which the following inequalities hold

$$\|\underline{K}-\underline{K}_1\| \leqslant c(\varepsilon)\,\delta(\underline{K},\underline{K}_1) \leqslant \|\underline{K}-\underline{K}_1\|,$$

where the left inequality holds provided $\delta(\underline{K},\underline{K}_1) < \varepsilon$.

Prop 3 implies that the gap and norm metrics are equivalent for stable systems. However, the gap metric is more general than the norm in as much as the gap is defined for unstable systems and the norm is not.

Prop 4 If $\underline{K}$ and $\underline{K}_1$ in $\underline{\mathbb{B}}_D$ have inverses in $\underline{\mathbb{B}}_D$, then $\delta(\underline{K},\underline{K}_1) = \delta(\underline{K}^{-1},\underline{K}_1^{-1})$.

Unlike the norm, the gap metric is preserved under inversion. This property is advantageous in the study of the feedback transformation $\underline{K} \mapsto (\underline{I}+\underline{K})^{-1}$.

The codimension of any subspace $\mathcal{M}$ of $\mathcal{H}$ is the dimension of the orthogonal complement of $\mathcal{M}$.

Prop 5 If the gap between two systems in $\mathbb{B}_D$ is small enough, then their stable subspaces must have equal codimension.

For frequency response systems $\underline{K}$ in $\mathbb{R}_o$, the codimension of St(K) is the number of poles of $\hat{K}(s)$ in $Re(s) > 0$. Prop 5 implies that two systems in $\mathbb{R}_o$ can not closely approximate each other if their numbers of $Re(s) > 0$ poles differ. If they do not differ, then a sufficient condition for close approximation is obtained as follows.

Let $p(\hat{K})$ denote the set of poles of $\hat{K}(s)$ in $Re(s) > 0$, and $d(\cdot,\cdot)$ the distance function between sets.

Prop 6 If $\underline{K} \in \mathbb{R}_o$, then there are constants $\varepsilon > 0$ and $\alpha > 0$, $\beta > 0$, for which the inequality

$$\delta(\underline{K},\underline{K}_1) \leqslant \beta \sup_{\omega} |\hat{K}(j\omega) - \hat{K}_1(j\omega)| + \alpha\, d[p(\hat{K}), p(\hat{K}_1)]$$

holds for any $\underline{K}_1$ in $\mathbb{R}_o$ which has the same number of $Re(s)>0$ poles as $\underline{K}$, and which satisfies the constraint $d[p(\hat{K}),p(\hat{K}_1)]<\varepsilon$.

Prop 6 suggests that in order to obtain a good approximation in gap to a system in $\mathbb{R}_o$ it is enough to obtain good estimates of its $j\omega$-axis frequency response, and of the locations of all its rhp poles. We shall show that such an approximation is also good for feedback purposes.

4. CONTINUITY OF FEEDBACK UNDER THE GAP METRIC

Consider the problem of approximating a feedback system $\underline{K}(\underline{I}+\underline{K})^{-1}$ to within a tolerance ε in $\mathbb{B}$, given that the open-loop system $\underline{K}$ is in $\mathbb{A}$. If $\underline{K}_1$ in $\mathbb{A}$ is some nominal value of the open-loop system for which the feedback system $\underline{K}_1(\underline{I}+\underline{K}_1)^{-1}$ is (stable) in $\mathbb{B}$, our problem is to ensure that $\underline{K}(\underline{I}+\underline{K})^{-1}$ remains in $\mathbb{B}$, and that the inequality

$$\|\underline{K}(\underline{I} + \underline{K})^{-1} - \underline{K}_1(\underline{I} + \underline{K}_1)^{-1}\| < \varepsilon \qquad (1a)$$

is satisfied. As $\underline{K}(\underline{I}+\underline{K})^{-1} = \underline{I} - (\underline{I}+\underline{K})^{-1}$, (1a) is equivalent to the inequality

$$\|(\underline{I} + \underline{K})^{-1} - (\underline{I} + \underline{K}_1)^{-1}\| < \varepsilon \qquad (1b)$$

we would like to find a way of specifying a tolerance on open-loop plants which will ensure that (1) is satisfied. In mathematical terms we seek a distance function or "metric" $m(\underline{K},\underline{K}_1)$ between pairs of plants $(\underline{K},\underline{K}_1)$ with the following property: For some $\delta>0$, if $m(\underline{K},\underline{K}_1)<\delta$ then $\underline{K}$ is stable and (1) holds; and conversely. The following theorem shows that the gap metric has the desired property.

Suppose that $\underline{K}_1 \in \mathbb{B}_D$ and that $(\underline{I}+\underline{K}_1)^{-1}$ is (stable) in $\mathbb{B}$.

Theorem 1 (a) There is a constant $\varepsilon > 0$ such that if $(\underline{I}+\underline{K})^{-1}$ satisfies

(1), then $\underline{K}$ is in $\mathbb{B}_D$ and there is a constant $\mu(\underline{K}_1,\varepsilon) > 0$ with the property that $\delta(\underline{K}.\underline{K}_1) < \mu$.

(b) Conversely, for any $\varepsilon > 0$ there is a constant $\eta(\underline{K}_1,\varepsilon) > 0$ such that if $\underline{K}$ is in $\mathbb{B}_D$ and $\delta(\underline{K},\underline{K}_1) < \eta$, then $(\underline{I}+\underline{K})^{-1}$ is (stable) in $\mathbb{B}$ and (1) is satisfied.

Remarks Theorem (1a) implies that the only tolerable uncertainties in $\underline{K}$ are those that are small in gap, at least if ε is small enough. Conversely, Theorem (1b) implies that uncertainties in $\underline{K}$ that are sufficiently small in gap are tolerable. Therefore, the gap metric precisely characterizes the tolerable small uncertainties. Furthermore, the gap metric is unique in this respect, modulo certain equivalences, as the following corollary shows.

Let $\mathbb{E}: \mathbb{B}_D \to \mathbb{B}$ be the mapping that takes open-loop systems $\underline{K}$ (in some subset of $\mathbb{B}_D$) into stable systems $(\underline{I}+\underline{K})^{-1}$ in $\mathbb{B}$. A metric $m(\cdot,\cdot)$ on $\mathbb{B}_D$ is called continuously robust for $\mathbb{E}$ if it has the following property: Given any $\underline{K}_1$ in $\mathbb{B}_D$ such that $(\underline{I}+K_1)^{-1}$ is in $\mathbb{B}$, and any $\varepsilon > 0$, there is a $\mu > 0$ with the property that if $m(\underline{K},\underline{K}_1) < \mu$ then $(\underline{I}+\underline{K})^{-1}$ is stable and (1) holds. Two metrics m_1 and m_2 are called equivalent if for any $\underline{K}_1$ in $\mathbb{B}_D$ there are positive constants μ_1,μ_2 c_1,c_2 for which the following inequalities hold

$$m_1(\underline{K},\underline{K}_1) \leqslant c_2\ m_2(\underline{K},\underline{K}_1) \leqslant \underline{c}_1\ m_1(\underline{K},\underline{K}_1)$$

providing that $m_1(\underline{K},\underline{K}_1) < \mu_1$ in the case of the right inequality, or $m_2(\underline{K},\underline{K}_1) < \mu_2$ in the case of the left one.

Corollary Every continuously robust metric for $\mathbb{E}$ is equivalent to the gap metric.

Remark For frequency response systems in $\mathbb{R}_o$, Theorem 1 implies that no error can be tolerated in the number of open-loop rhp poles. However, small uncertainties in the open-loop poles positions and frequency response can be tolerated.

Since the number of rhp poles can not be deduced with certainty from purely input-output measurements, Theorem 1 suggests that unstable systems can not be stabilized by feedback without some knowledge of their internal structure (See, e.g., the Appendix of [2]).

REFERENCES

[1] A. El-Sakkary, The Gap Metric for Unstable Systems, Ph.D. thesis, McGill University, Department of Electrical Engineering, 1980.

[2] G. Zames, "Feedback and Optimal Sensitivity: Model Reference Transformations, Weighted Seminorms, and Approximate Inverses". McGill University, Department of Electrical Engineering Report, 1979; summarised in Proc. 17th Allerton Conf., Oct. 1979, pp. 744-752.

AN APPLICATION OF TOTAL SYNTHESIS TO ROBUST COUPLED DESIGN*

M.K. SAIN, R.M. SCHAFER, AND K.P. DUDEK
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

Total synthesis techniques for the control of linear multivariable systems are module theoretic methods aimed at bottom line design, wherein the designer selects at the outset the control action and system response characteristics which will be attained. This selection is made from the class having associated with it the possibility of an internally stable feedback realization. In practice, however, such realizations are normally carried out with decentralized dynamics---or fixed structure. Moreover, they must be achieved in such a way that plant parameter variation does not materially inhibit the nominal filter responses. This paper reports on the extension of the Total Synthesis Problem (TSP) concept to robust feedback realizations of fixed structure. An example from turbojet engine data illustrates the concepts.

INTRODUCTION

In the following discussion, the object of control is understood to satisfy the state equations

$$\dot{x} = Ax + Bu \tag{1a}$$

$$y = Cx + Du , \tag{1b}$$

for x, u, and y members of the n, m, and p dimensional real vector spaces X, U, and Y, respectively. This brief exposition will assume that control goals include a command/response relationship between an exogenous signal r, which takes its values in a p dimensional real vector space R, and the pair (u,y).

Consider now the expression of such a relationship in the frequency domain. Denote the field of real numbers by $\mathbb{R}$, the principal ideal domain ring generated from $\mathbb{R}$---considered as a commutative ring---and the indeterminate element s by $\mathbb{R}[s]$, and the quotient field of $\mathbb{R}[s]$ by $\mathbb{R}(s)$. Then the constructions [1]

$$U[s] = U \otimes_{\mathbb{R}} \mathbb{R}[s] , \tag{2a}$$

$$X[s] = X \otimes_{\mathbb{R}} \mathbb{R}[s] , \tag{2b}$$

$$Y[s] = Y \otimes_{\mathbb{R}} \mathbb{R}[s] , \tag{2c}$$

$$R[s] = R \otimes_{\mathbb{R}} \mathbb{R}[s] \tag{2d}$$

establish free $\mathbb{R}[s]$-modules U[s], X[s], Y[s], and R[s], respectively. In turn, these imbed naturally into the $\mathbb{R}(s)$-vector spaces

$$U(s) = U \otimes_{\mathbb{R}} \mathbb{R}(s) , \tag{3a}$$

---

*This work was supported by the Office of Naval Research under Contract N00014-79-C-0475.

$$X(s) = X \otimes_{\mathbb{R}} \mathbb{R}(s) \ , \tag{3b}$$

$$Y(s) = Y \otimes_{\mathbb{R}} \mathbb{R}(s) \ , \tag{3c}$$

$$R(s) = R \otimes_{\mathbb{R}} \mathbb{R}(s) \ , \tag{3d}$$

in the manner

$$U[s] \subset U(s) \ , \tag{4a}$$

$$X[s] \subset X(s) \ , \tag{4b}$$

$$Y[s] \subset Y(s) \ , \tag{4c}$$

$$R[s] \subset R(s) \ . \tag{4d}$$

In this context, the plant appears as a morphism

$$P : U(s) \to Y(s) \ . \tag{5}$$

It is further assumed that there exists a morphism

$$M : R(s) \to U(s) \ , \tag{6}$$

which embodies the command/control action relationship, and that there exists a morphism

$$T : R(s) \to Y(s) \ , \tag{7}$$

which characterizes the command/system response relationship. Broadly speaking, the assumptions reflected in (6) and (7) mean that the discussion is in focus on linear feedback control systems.

Practical design and computation can be carried out through a process of basis selection in U, X, Y, and R. Understood as elements of U[s], X[s], Y[s], and R[s], these basis vectors become free generators of their associated $\mathbb{R}[s]$-modules. In like manner, understood as elements of U(s), X(s), Y(s), and R(s), they comprise a basis for their associated $\mathbb{R}(s)$-vector spaces as well. Denote by [L] the matrix of a morphism

$$L : V_1 \to V_2 \tag{8}$$

of F-vector spaces V_i, i = 1,2. Then (5), (6), and (7) have their counterparts

$$[P] : \mathbb{R}(s)^m \to \mathbb{R}(s)^p \ , \tag{9}$$

$$[M] : \mathbb{R}(s)^p \to \mathbb{R}(s)^m \ , \tag{10}$$

$$[T] : \mathbb{R}(s)^p \to \mathbb{R}(s)^p \ . \tag{11}$$

These ideas and notations are now applied to define and elaborate the idea of total synthesis.

THE TOTAL SYNTHESIS PROBLEM

The idea of total synthesis revolves around the commutative diagram of Figure 1, which points out the basic feature that any pair (M,T) brought about by a feedback control system must coexist with the plant or object of

$$\begin{tikzcd} R(s) \arrow[d, "M"'] \arrow[dr, "T"] & \\ U(s) \arrow[r, "P"'] & Y(s) \end{tikzcd}$$

Figure 1.

$$\begin{tikzcd} R(s) \arrow[d, dashed, "M"'] \arrow[dr, dashed, "T"] & \\ U(s) \arrow[r, "P"'] & Y(s) \end{tikzcd}$$

Figure 2.

control. In other words, with

$$P = C(s1_X - A)^{-1} B + D \ , \tag{12}$$

the pair (M,T) must satisfy

$$T = P \circ M \ . \tag{13}$$

The Total Synthesis Problem (TSP) is brought about through a study of the possible pairs (M,T) which can satisfy (13) for a given P. TSP may be conceptualized as follows. Denote by

$$\hom\,(R(s),\, Y(s)) \tag{14}$$

The $\mathbb{R}(s)$-vector space of morphisms with domain $R(s)$ and codomain $Y(s)$. In like manner, establish

$$\hom\,(R(s),\, U(s)) \ , \tag{15}$$

with the alternative codomain $U(s)$. Then the action

$$(M,T) \mapsto P \circ M - T \tag{16}$$

establishes a morphism

$$\hom\,(R(s),\, U(s)) \times \hom\,(R(s),\, Y(s)) \rightarrow \hom\,(R(s),\, Y(s)) \ . \tag{17}$$

Denote this morphism by F; then $\operatorname{Ker} F$ is an appropriate setting within which to begin a study of TSP.

It is important to realize at the outset, however, that the rational subspace $\operatorname{Ker} F$ is much too capacious a place in general within which to design control systems. Indeed, arbitrary pairs (M,T) taken from $\operatorname{Ker} F$ may not be feedback realizable in an internally stable way. It is easy to see why this is the case. Suppose that $[M]$ contained unstable elements of $\mathbb{R}(s)$. Then a command r would generate unstable control action u, a situation which is at odds with the notion of internal stability. Moreover, internal stability would also require that $[T]$ have stable elements from $\mathbb{R}(s)$ as well. Accordingly, the pair $([M], [T])$ must have its $\mathbb{R}(s)$-elements in the stable subring of $\mathbb{R}(s)$. Beyond this natural condition, it is clearly desirable from the practical point of view to require each entry in $[M]$ and in $[T]$ to be proper. Failure to do so would create unacceptable transients in response to fast commands.

When D is the zero morphism, Bengtsson [2] established in 1977 the results which could be used to show that ℝ(s)-elements in [M] and [T] being stable and proper was both necessary and sufficient for the pair (M, T) to admit an internally stable feedback realization. However, Bengtsson's working assumptions involved centralized controller dynamics, whereas many applications require the distribution of dynamics in the controller to more than one site. Stated in other words, the Bengtsson configuration places little constraint upon the connections within the controller, whereas many applications have such constraints.

If connection constraints within the controller make it useful to distribute the controller dynamics to more than one location, then the stable and proper conditions on ([M], [T]), while still necessary, may not be sufficient for internal stability.

COMPUTATIONAL ASPECTS

As discussed in the section preceding, TSP studies of possible pairs (M,T) are located generally within Ker F. In particular, however, conditions of internally stable feedback realizability require that the study be localized to pairs (M,T) whose matrices ([M],[T]) are proper and stable. Because the subring of proper and stable elements in ℝ(s) fails to achieve the status of a field---the multiplicative inverse of a stable transfer function need not be stable---it is appropriate to examine Ker F from a module theoretic, as opposed to a vector space theoretic, point of view. A practical approach to computing the stable and proper part of Ker F is to suppress, temporarily, the properness condition and to calculate a basis for the stable part of Ker F. Because transfer functions without denominators may be regarded as stable in this sense, polynomials can serve as such basis elements. Once the polynomials are obtained, properness can be restored by dividing basis elements with Hurwitz polynomials of appropriate degree, such operations being just basis transformations within the stable kernel. The idea can be seen in the following first step of the turbojet example.

Turbojet Example: Step 1

Consider the single spool turbojet engine model due to Skira and DeHoff [3]. For this model, the plant state description is given by

$$[A] = \begin{bmatrix} -1 & -.5 \\ 10 & -5 \end{bmatrix}; \quad [B] = \begin{bmatrix} 2 & .5 \\ 0 & -1 \end{bmatrix};$$
$$[C] = \begin{bmatrix} .1 & 18 \\ .06 & 0 \end{bmatrix}; \quad [D] = \begin{bmatrix} 0 & 0 \\ .46 & 0 \end{bmatrix}. \tag{18}$$

Then a brief calculation establishes that

$$[P] = \frac{\begin{bmatrix} 36.1(.000554s + 1) & 7.23(-.248s + 1) \\ .52(.0884s^2 + .554s + 1) & .018(.1667s + 1) \end{bmatrix}}{(.1s^2 + .6s + 1)}. \tag{19}$$

Let

$$d_P(s) = .1s^2 + .6s + 1\ , \tag{20}$$

and denote by N_P the numerator matrix in (19). The kernel problem can be understood in the manner

$$[N_P \,\vdots\, - d_P(s) I] \left[\begin{array}{c} N_M \\ \hline N_T \end{array}\right] = 0\ , \tag{21}$$

where the representations

$$[M] = \frac{N_M}{d_M(s)}\ , \tag{22}$$

$$[T] = \frac{N_T}{d_T(s)}\ , \tag{23}$$

have been employed. Columns of the matrix

$$\left[\begin{array}{c} N_M \\ \hline N_T \end{array}\right] \tag{24}$$

are computed so that they constitute a basis for the stable part of Ker F. In this example, they were found to be

$$\begin{bmatrix} -2.5 & 2.17s - 8.70 \\ s + 10 & 43.5 \\ -17.95 & .435 \\ -1.12 & s - 3.74 \end{bmatrix}, \tag{25}$$

so that a base choice for N_M and N_T is

$$N_M = \begin{bmatrix} -2.5 & 2.17s - 8.70 \\ s + 10 & 43.5 \end{bmatrix}, \tag{26}$$

$$N_T = \begin{bmatrix} -17.95 & .435 \\ -1.12 & s - 3.74 \end{bmatrix}. \tag{27}$$

However, it would be rare indeed if the choices (26) and (27) turned out to be physically satisfactory, inasmuch as they represent an arbitrary choice of basis. What is needed next in the design process is a manipulation of the basis (25) so as to move toward a desirable practical design. One approach to this goal is to adjust the zeros and gains of [P] so as to reach an alternate pattern. Notice that the given gain matrix of [P] is just

$$[P]\big|_{s=0} = \begin{bmatrix} 36.1 & 7.23 \\ .52 & .018 \end{bmatrix}. \tag{28}$$

The signs and magnitudes of the individual entries in (28) may be taken as an indication of the natural propensities of the plant. While changes in the pattern of (28) may reflect interesting design goals, it is necessary to bear in mind that gross changes may tend to invalidate the original plant data or may result in the plant operating locally in a manner which is not in keeping with its global features. Notice also that the 11-element of [P] has a rather large left half plane zero, and that the 12-element of

[P] has a right half plane zero. As an example design strategy, suppose that it is chosen to

i) move the zero of the 11-element farther out in the left half plane;

ii) reduce the gain in the 21-element;

iii) remove the zero of the 12-element from the right half plane.

A least squares procedure can be applied in order to pick polynomic entries in Ker F which are in the aggregate as close as possible to matrices N_T which would have the three properties listed above. The result of one such calculation is given by

$$\begin{bmatrix} .0291s + .0223 & -.289s + 2.3 \\ -2.79 \times 10^{-5} s^2 - .0557s + .0271 & -.0531s^2 - .775s - 10.9 \\ 5.00 \times 10^{-4} s + 1 & .953s + 4.31 \\ .0134s + .0121 & -.135s + 1 \end{bmatrix}, \qquad (29)$$

where N_M appears in the top two rows and N_T in the bottom two rows. Here

$$N_T|_{s=0} = \begin{bmatrix} 1 & 4.31 \\ .0121 & 1 \end{bmatrix}; \qquad (30)$$

under the choice

$$d_T(s) = d_M(s) = (.2s+1)(.04s+1) \ , \qquad (31)$$

(30) may be compared with (28). The sign pattern in the gain matrix has been maintained, with some readjustment of relative gain magnitudes in the second channel, as well as near normalization in the first channel. The zero of the 11-element has moved to -2,000 from its original position; the gain in the 21-element has decreased from .52 to .0121; and the zero in the 12-element has migrated from 4.03 in the right half plane to -4.52 in the left half plane. As is usually the case, this adjustment must be traded off against other changes which occur consequentially. Of particular importance is the creation of a right half plane zero in the 22-element. Also worthy of mention is the use of second degree elements in the kernel; these are seen in the upper two rows of (29). Second degree elements were employed in order to achieve more satisfactorily the three design goals elaborated above. The consequence is greater complexity in the controller.

Remark 1

The use of (21) to define the kernel problem permits interpretation of the calculation in terms of free $\mathbb{R}[s]$-modules along the lines of (2). This way of thinking has been presented in [4]. However, the infinite precision algorithms discussed in [4] are not useful in the case of finite precision [5]. Instead, it is convenient to retain the free-modular viewpoint while constructing a minimal (sum of column degrees) basis with the aid of ideas of Foster [6] or of Kung and Kailath (see [7]).

Remark 2

Second degree elements, as appearing in (29), may connote one of two events. If only free $\mathbb{R}[s]$-modular basis changing isomorphisms are used to construct (29) from (25), then the solution indicates a non-minimal (sum of column degrees) basis. If (25) is adjusted in the larger class of monic, rank two morphisms to reach (29), then attention in Ker F has been specialized to a submodule. The latter case is ordinarily to be avoided because it increases the number of closed loop system multivariable zeros beyond the required minimum quantity present in (25).

ROBUST FEEDBACK REALIZATION

Suppose next that feedback realization of the pair (M,T) is to be achieved within the fixed configuration of Figure 3. Regard T, as determined in the preceding discussion---cf. (29) and (31)---as given; and recall that the plant of the turbojet example is stable. Then G and H : $Y(s) \rightarrow R(s)$ will be chosen so as to realize the pair (M,T) and to reach a given level of comparison sensitivity to inaccuracies in the plant. According to the results of Cruz and Perkins [8], it is a straightforward derivation to show that $S : Y(s) \rightarrow Y(s)$, the comparison sensitivity operator, satisfies the pair of equations

$$S \circ P \circ G = T , \tag{32}$$

$$T \circ H = 1_{Y(s)} - S . \tag{33}$$

Because S is an isomorphism, (32) can be expressed

$$P \circ G = S^{-1} \circ T . \tag{34}$$

For this brief treatment, suppose that P is monic and that

$$[S] = \frac{n(s)}{d(s)} I , \tag{35}$$

with $(n(s),d(s)) \in \mathbb{R}[s]^2$ and $d(s) \neq 0$. Then

$$G = S^{-1} \circ M , \tag{36}$$

which is unstable only if n(s) is unstable. Moreover, (33) may be regarded as an equation for pairs (H,S), in which case it has an especially interesting particular solution

$$(0, 1_{Y(s)}) . \tag{37}$$

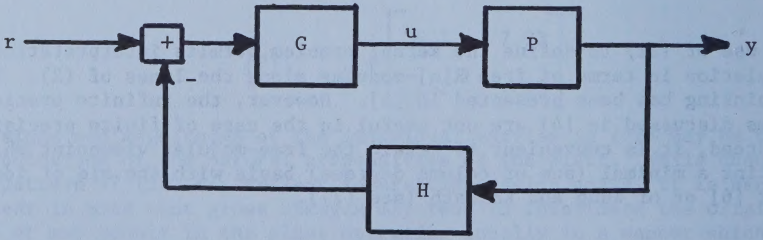


Figure 3. Fixed Structure Feedback Realization.

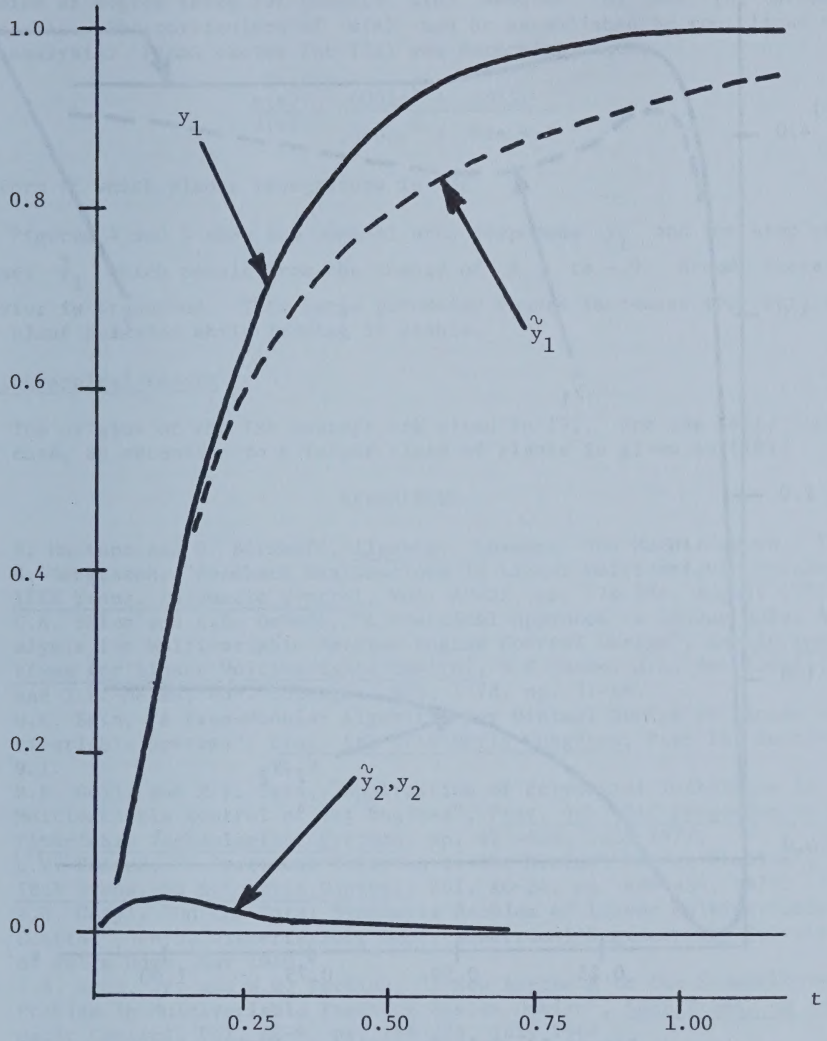


Figure 4. Unit Step in First Channel.

The complementary solutions can be found from another kernel problem

$$[T \mid I]\left[\begin{array}{c} H \\ \hdashline S \end{array}\right] = 0 \; ; \qquad (38)$$

because [S] must be stable in order to ensure internal loop stability, and because instabilities in [H] would be undesirable from the viewpoint of multivariable zeros in the closed loop, the same module theoretic and computational ideas described above for (M,T) pairs may be applied here to (H,S) pairs. To accommodate the sort of assumption (35), softwares of the Foster or Kung-Kailath type can be appropriately modified.

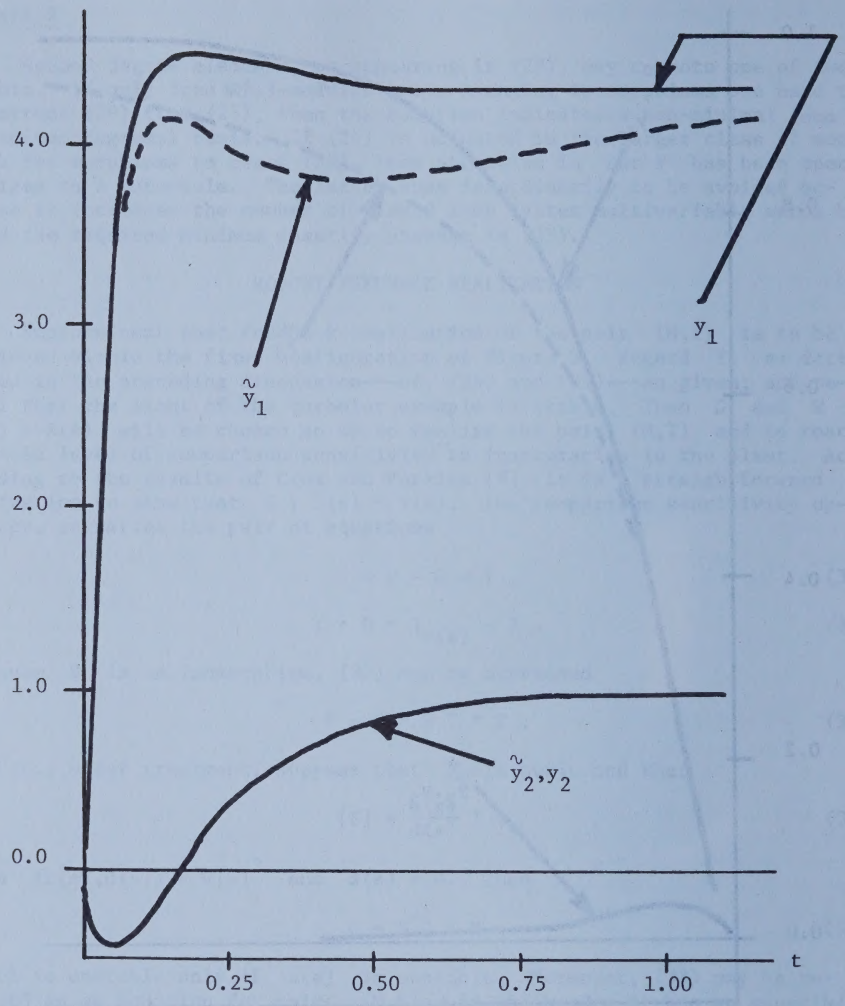


Figure 5. Unit Step in Second Channel.

Taken all together, [T] stable, [G] stable, [P] stable, and [H] stable preclude internal instabilities.

Turbojet Example: Step 2

Calculation of a basis for stable complementary solutions in (38), under the constraint (35), yields the result

$$\begin{bmatrix} N_H \\ \hdashline \tilde{n}(s)I \end{bmatrix}, \tag{39}$$

where $\tilde{n}(s)$ has its zeros equal to the multivariable zeros of the [T], and where N_H is of degree three---a consequence of the constraint (35).

A choice of degree three for Hurwitz d(s) assures [H] and [S] proper and stable. The particulars of d(s) may be established by root-locus or Bode analysis. Final choice for (35) was determined to be

$$\frac{n(s)}{d(s)} = \frac{.0001s^2 + .2855s}{.0001s^2 + .02s + 1} \tag{40}$$

the form of which places integrators in G.

Figures 4 and 5 show the nominal step responses y_i and the step responses $\tilde{y}_i$ which result from the change of A_{12} to -.9. Steady state behavior is preserved. This large parameter change increases the oscillatory plant behavior while leaving it stable.

Bibliographical Remark

The origins of the TSP concept are cited in [9]. For the unity feedback case, an extension to a larger class of plants is given in [10].

REFERENCES

1. S. MacLane and G. Birkhoff, Algebra. London: The MacMillan Co., 1967.
2. G. Bengtsson, "Feedback Realizations in Linear Multivariable Systems", IEEE Trans. Automatic Control, Vol. AC-22, pp. 576-585, August 1977.
3. C.A. Skira and R.L. DeHoff, "A Practical Approach to Linear Model Analysis for Multivariable Turbine Engine Control Design", in Alternatives for Linear Multivariable Control, M.K. Sain, J.L. Peczkowski, and J.L. Melsa, eds. Chicago: NEC, 1978, pp. 31-46.
4. M.K. Sain, "A Free-Modular Algorithm for Minimal Design of Linear Multivariable Systems", Proc. 6th IFAC World Congress, Part IB, Section 9.1.
5. R.R. Gejji and M.K. Sain, "Application of Polynomial Techniques to Multivariable Control of Jet Engines", Proc. 4th IFAC Symposium on Multivariable Technological Systems, pp. 421-429, July 1977.
6. L.V. Foster, "A Practical Solution to the Minimal Design Problem", IEEE Trans. on Automatic Control, Vol. AC-24, pp. 449-454, 1979.
7. R.R. Gejji, "On the Total Synthesis Problem of Linear Multivariable Control", Ph.D. Dissertation, Dept. Electrical Engineering, University of Notre Dame, May 1980.
8. J.B. Cruz, Jr. and W.R. Perkins, "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design", IEEE Trans. on Automatic Control, Vol. AC-9, pp. 216-223, July 1964.
9. M.K. Sain and A. Ma, "Multivariable Synthesis with Reduced Comparison Sensitivity", Proc. 1980 Joint Automatic Control Conference, Part WP8-B.
10. C.A. Desoer and M.J. Chen, "Algebraic Design of Unity Feedback Systems with Stable Plant", Proc. 1980 Joint Automatic Control Conference, Part TA8-E.

GAIN AND PHASE MARGINS FOR LINEAR MULTIVARIABLE FEEDBACK SYSTEMS

IAN POSTLETHWAITE
General Electric Company, Schenectady, NY 12345, and
Cambridge University, Cambridge, England.

Abstract. The relative stability of a linear, time-invariant, single-input, single-output, feedback system is determinable from the closeness of the Nyquist diagram to the so-called critical point. In this paper a generalization of this technique is developed in order to characterize the relative stability of a multi-input, multi-output, feedback system.

1. INTRODUCTION

To a control systems design engineer, control system and feedback system almost go hand in glove, because of the many desirable properties associated with a feedback structure. For example, the effects of model uncertainty, parameter variations, external disturbances, and nonlinearities on a desired input-output relationship can all be reduced by feedback; [1], [2], and [3]. In theory, all these nice properties are obtained if the loop gains are large enough over the frequency range of interest. In multi-input, multi-output (multivariable) systems large loop gains are also a sufficient condition for low interaction, and so a general feedback rule might be: for good performance use large loop gains.

There are, of course, limitations to the amount of gain that can be used in a practical control system. For instance, large gains result in large input signals which may cause saturation or at least take the system model out of its linear operating region. Also, large loop gains correspond to large energy requirements which may breach the problem specification. Finally, and perhaps most important, feedback, if applied incorrectly, can cause instability, and so one has to be very careful that the use of large loop gains does not seriously deteriorate the closed-loop stability margin.

In the single-input, single-output case the degree of stability of the feedback system can be determined using the classical frequency-response approach due to Nyquist [4], and the associated gain and phase margins. Although there is a generalization of Nyquist's stability criterion for multivariable systems, [5], [6] and [7], in terms of the eigenloci of the open-loop frequency-response matrix, the associated gain and phase margins do not reliably characterize the robustness of the closed-loop stability property. This has been highlighted by Doyle [8], who has further shown how to characterize the robustness of the closed-loop stability property in terms of the maximum singular value (spectral norm) of an appropriate frequency-response matrix; see also [9] and [10]. This robustness result, which can be viewed as a generalization of gain margin, can be derived from a simple application of the small gain theorem, [11], [12] and [13], and as such is generally a conservative measure for relative stability. In this paper a small phase theorem is developed which generalizes the concept of phase margin. When used in conjunction with the the small gain theorem, the small phase theorem allows a larger class of perturbations of the open-loop system to be identified for which the closed-loop system remains stable than when singular values alone are used.

The phase information, on which the small phase theorem is based, stems from the polar decomposition of a complex matrix [14], which is defined as follows. Analagous to the polar form of a complex number, a complex matrix T, can be represented in the forms

$$T = UH_R \qquad (1.1)$$

$$T = H_L U \qquad (1.2)$$

where U is unitary and H_R, H_L are positive-semidefinite Hermitian matrices. H_R, H_L, which are often called the right and left moduli of T, are uniquely determined as $\sqrt{T^*T}$, $\sqrt{TT^*}$ respectively, where * denotes complex conjugate transpose, and U is uniquely defined via (1.1) or (1.2) when T is nonsingular. Representations (1.1) and (1.2) are called polar decompositions of T, and are easily determined from the singular value decomposition of T, [16], for which there exists well tested software. If T has the singular value decomposition

$$T = X\Sigma V^* \qquad X,V \text{ unitary; } \Sigma \text{ diagonal and } \geq 0 \tag{1.3}$$

then
$$T = (XV^*)(V\Sigma V^*) = UH_R \tag{1.4}$$

and
$$T = (X\Sigma X^*)(XV^*) = H_L U \; . \tag{1.5}$$

It will be useful in the context of this paper to think of T as a complex matrix obtained from a Laplace transfer function T(s) evaluated at some complex frequency $s=j\omega$.

From the polar decompositions the following key quantities are defined:

(i) the characteristic gains of T are its eigenvalues;

(ii) the principal gains of T are the eigenvalues of the Hermitian part of its polar decomposition;

(iii) the principal phases of T are the arguments of the eigenvalues of the unitary part of its polar decomposition.

Note that the eigenvalues of T^*T are the same as those of TT^*, and so either decomposition may be used in (ii). Also note that principal gains are frequently used in the field of numerical analysis where they are referred to as singular values, and more commonly defined as the diagonal elements of Σ in (1.3).

In section 2, two important theorems are presented which relate the characteristic gains to the principal gains and phases of a complex matrix T. Then based on these theorems a sufficient Nyquist-type stability criterion is developed in section 3.

In section 4, it is shown how the sufficient Nyquist-type criterion can be used to characterize the robustness of the closed-loop stability property when the system is subjected to additive or multiplicative-type perturbations. The major new result of this section is theorem 4, the small phase theorem, which effectively gives limits to the arguments of the characteristic gains of a product of two matrices in terms of the principal gains and phases of the individual matrices. This result allows the prediction of closed-loop stability for a perturbation which has infinitely large gains over a frequency range, providing the condition number and principal phases of the perturbation satisfy certain requirements.

This invited paper is to all intent and purpose a summary of the major results that appeared in [17], and therefore it will contain no proofs.

2. RELATIONSHIPS BETWEEN THE CHARACTERISTIC GAINS AND THE PRINCIPAL GAINS AND PHASES OF A COMPLEX MATRIX

In this section two important theorems are presented which relate the characteristic gains to the principal gains and phases of a complex matrix T. The theorems are crucial to the stability criterion of the next section.

Theorem 1. The magnitudes of the characteristic gains of a complex matrix T are bounded above and below by the corresponding maximum and minimum principal gains.

Theorem 2. If the principal phases of a complex matrix T have a spread of less than π then the arguments of the characteristic gains of T are bounded above and below by the maximum and minimum principal phases.

3.A SUFFICIENT NYQUIST-TYPE STABILITY CRITERION

In the generalized Nyquist stability criterion [5],[6] and [7], the stability of the linear multivariable feedback configuration shown in figure 1, is determined from the characteristic gain loci ,which are a polar plot of the eigenvalues of the open-loop gain matrix $G(s)$, evaluated as s traverses the standard Nyquist D-contour. The actual criterion can be stated as follows:

> the closed-loop system is stable if,and only if ,the number of anti-clockwise encirclements of the critical point $-1 \div k$, by the characteristic gain loci, is equal to the number of open-loop unstable poles.

It is now shown how, using theorems 1 and 2, a sufficient Nyquist-type stability criterion can be obtained in terms of the principal gains and phases of $G(j\omega)$.

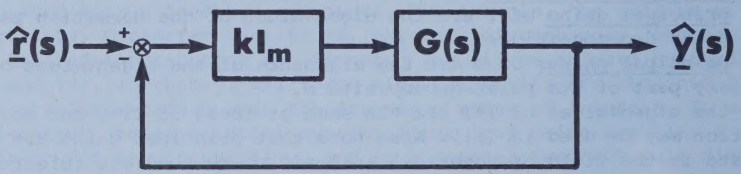


Figure 1.Multivariable feedback configuration

For an open-loop gain matrix $G(s)$ it is clear from theorems 1 and 2 that for any complex frequency $s=j\omega$ a curvilinear rectangle (segment of annular region) can be drawn from the maximum and minimum principal values, within which the corresponding m values of the characteristic gain loci (generalized Nyquist diagrams) lie; m is the number of inputs and outputs. Should the principal phases not form a convex cone, that is, have a spread greater than or equal to π, then the characteristic gains lie inside an annular region determined by the maximum and minimum principal gains. If these curvilinear rectangles and annular regions are constructed for values of s around the Nyquist D-contour[1] then a region will be outlined in the gain plane, called a principal region, within which the characteristic gain loci must lie; see figure 2. Application of the generalized Nyquist stability criterion to this region leads to the following stability criterion:

> the closed-loop system is stable if m times the number of anti-clockwise encirclements of the critical point $-1 \div k$, by the principal region, is equal to the number of open-loop unstable poles.

It also follows that:

> the closed-loop system is unstable if m times the number of anti-clockwise encirclements of the critical point $-1 \div k$, by the principal region, is not equal to the number of open-loop unstable poles.

When the critical point lies inside the principal region neither stability nor instability can be inferred.

Since the above are only sufficient conditions for stability, as straightforward stability tests they are inferior to the generalized Nyquist stability criterion. However, as highlighted by Doyle [8], the generalized Nyquist stability criterion only determines stability with respect to a single gain k common to all the loops, and as such does not reliably characterize the robustness of the stability property when the system is subjected to arbitrary perturbations. In the next section it is shown how this new sufficient Nyquist-type stability criterion can be used to characterize the robustness of a linear multivariable design when subjected to a matrix of linear, dynamic perturbations.

---

[1] As in the generalized Nyquist stability criterion, see[18] for example, the Nyquist D-contour has, in theory, ε-indentations around poles, zeros, and branch points which appear on the imaginary axis.

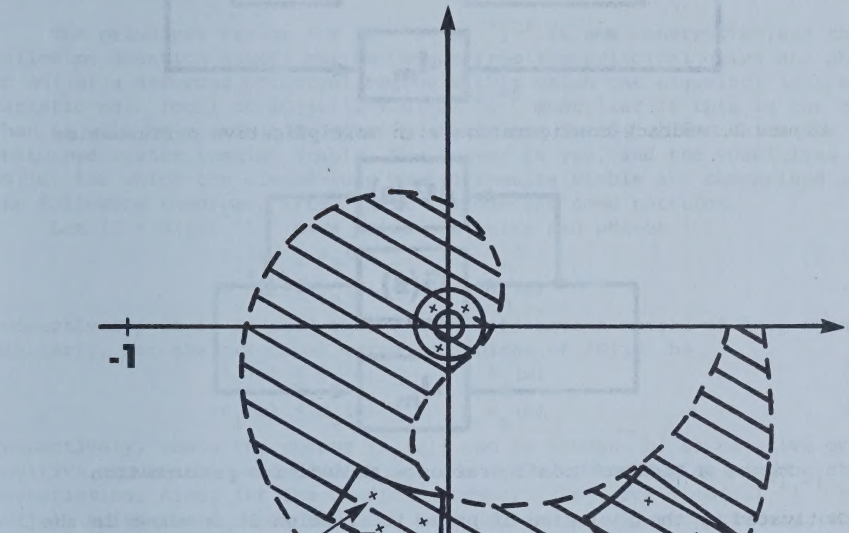


Figure 2. Principal region

4. ROBUST STABILITY

One of the major reasons for using feedback control, as opposed to open-loop control, is the presence of model uncertainties, and hence the desire to keep the system stable under large parameter variations. However, the mere presence of feedback is not sufficient to guarantee the robust stability of a closed-loop system, [1]. Research in this area has recently, [8], [9], and [10], centred around finding conditions for which a feedback system will remain stable when the open-loop gain matrix G(s) is subjected to a multiplicative or additive perturbation $\Delta G(s)$. It has been shown that the feedback configurations of figures 3 and 4, remain stable when $\Delta G(s)$ is stable and

$$||\Delta G(j\omega)|| < \frac{1}{||[I + G(j\omega)^{-1}]^{-1}||} \qquad \text{for all } \omega \qquad (4.1)$$

in the multiplicative case[2], and

$$||\Delta G(j\omega)|| < \frac{1}{||[I + G(j\omega)]^{-1}||} \quad \text{for all } \omega \qquad (4.2)$$

in the additive case[2]. If the spectral norm $||.||_2$ is taken, then the robustness of each configuration is charaterized by the maximum principal gain (singular value) of the appropriate frequency response-matrix.

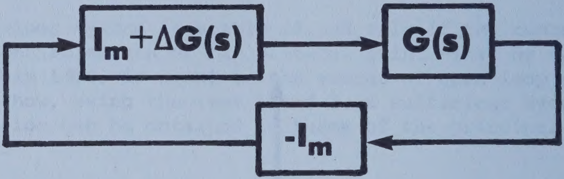


Figure 3.Feedback configuration with multiplicative perturbation

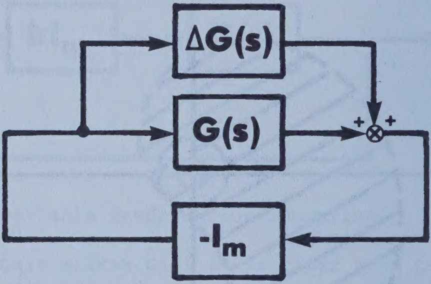


Figure 4.Feedback configuration with additive perturbation

Motivated by the inclusion of phase information it is shown in the following subsection how a larger class of perturbations can be identified for which the closed-loop system remains stable than when singular values alone are used.

4.1 Gain and phase margins

To use the new stability criterion in the context of robustness firstly rearrange figures 3 and 4, to those in figures 5 and 6, respectively, and make the observation that the system in figure 3 is stable if, and only if, the system of figure 5 is stable, and likewise for figures 4 and 6. The arguments that follow are essentially the same for both the multiplicative and additive cases, and so for simplicity we will consider only the multiplicative case.

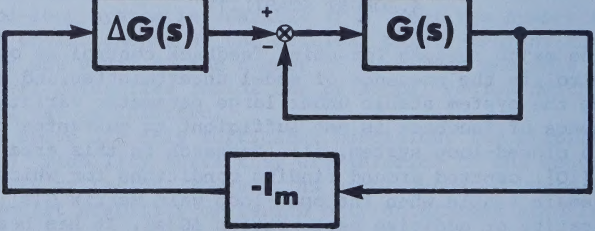


Figure 5.Rearranged feedback configuration with multiplicative perturbation

---

[2] These relationships are most easily derived from a simple application of the small gain theorem,[11],[12]and[13],after first rearranging the diagrams.

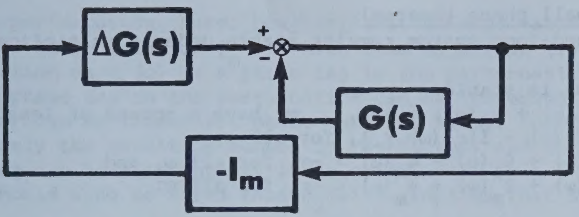


Figure 6. Rearranged feedback configuration with additive perturbation

The principal region for $[I + G(j\omega)^{-1}]^{-1}$ is now constructed, and the following question asked: can we deduce from the principal gains and phases of $\Delta G(j\omega)$ a deformed principal region within which the eigenloci (characteristic gain loci) of $\Delta G(j\omega)[I + G(j\omega)^{-1}]^{-1}$ must lie? If this is the case then it will be possible to specify conditions on $\Delta G(j\omega)$ for which the perturbed system remains stable. The answer is yes, and the conditions on $\Delta G(j\omega)$ for which the closed-loop system remains stable are summarised in the following theorems, after first introducing some notation.

Let $[I + G(j\omega)^{-1}]^{-1}$ have principal gains and phases

$$\alpha_1(\omega) \leq \alpha_2(\omega) \leq \ldots \leq \alpha_m(\omega) \tag{4.3}$$

$$\theta_1(\omega) \leq \theta_2(\omega) \leq \ldots \leq \theta_m(\omega) \tag{4.4}$$

respectively; it is assumed that the $\{\theta_i(\omega)\}$ have a spread of less than π. Similarly, let the principal gains and phases of $\Delta G(j\omega)$ be

$$\delta_1(\omega) \leq \delta_2(\omega) \leq \ldots \leq \delta_m(\omega) \tag{4.5}$$

$$\varepsilon_1(\omega) \leq \varepsilon_2(\omega) \leq \ldots \leq \varepsilon_m(\omega) \tag{4.6}$$

respectively, where the phases $\{\varepsilon_i(\omega)\}$ can be thought of as negative or positive arguments corresponding to unmodelled phase lag or advance characteristics. Also, let the condition numbers [16], for $[I+G(j\omega)^{-1}]^{-1}$ and $\Delta G(j\omega)$, using the ℓ_2-induced norm, be $c_1(\omega)$ and $c_2(\omega)$, respectively, so that

$$c_1(\omega) \triangleq \frac{\alpha_m(\omega)}{\alpha_1(\omega)} \quad \text{and} \quad c_2(\omega) \triangleq \frac{\delta_m(\omega)}{\delta_1(\omega)} \tag{4.7}$$

and also define

$$\psi_m(\omega) \triangleq \tan^{-1}\left[\frac{[c_1(\omega)-1]c_2(\omega)}{1-[c_1(\omega)-1]c_2(\omega)}\right] \tag{4.8}$$

which will be referred to as a phase modifier.

Theorem 3. (Small gain theorem)

The closed-loop system remains stable under a multiplicative perturbation $\Delta G(s)$, if

(i) $\Delta G(s)$ is stable, and

(ii) $\delta_m(\omega)\alpha_m(\omega) < 1$, for all ω.

This is exactly the small gain theorem introduced by Sandberg [11], and Zames [12], for a more general class of systems. The result is often conservative especially in our case where we are considering linear systems, but the conservativeness can be reduced if we consider the principal phases as described in the next theorem.

Theorem 4. (Small phase theorem)

The closed-loop system remains stable under a multiplicative perturbation $\Delta G(s)$, if

(i) $\Delta G(s)$ is stable,

(ii) $\{\theta_i(\omega) + \varepsilon_j(\omega): i,j=1,\ldots,m\}$ have a spread of less than π, for all ω,

(iii) $[c_1(\omega) - 1]c_2(\omega) < 1$, for all ω,

(iv) $\varepsilon_1(\omega) + \theta_1(\omega) - \psi_m(\omega) > -\pi$, for all ω, and

(v) $\varepsilon_m(\omega) + \theta_m(\omega) + \psi_m(\omega) < \pi$, for all ω.

Note that conditions (iv) and (v) are analagous to the phase margin concept used in the analysis of single-input, single-output feedback systems, modulo the phase modification $\psi_m(\omega)$.

In general, theorem 4 is more applicable at low frequencies when $\theta_1(\omega)$ is a small lag. At higher frequencies when $\theta_1(\omega)$ approaches $-\pi$, and the principal region is in the vicinity of the "-1" point, then theorem 3 is more useful. Consequently, for a more practical characterization of the robustness of the stability property over the whole frequency range, theorems 3 and 4 are combined, as in theorem 5 below.

Theorem 5.

The closed-loop system remains stable under a multiplicative perturbation $\Delta G(s)$, if for some frequency ω_b,

(i) the conditions of theorem 4 are satisfied in the frequency range $[0,\omega_b]$, and

(ii) the conditions of theorem 3 are satisfied in the frequency range $[\omega_b,\infty]$.

The proofs of theorems 3 and 4 are given in [17], but essentially they are based on the following ideas. In theorem 3, the moduli of the characteristic gain loci are restricted to values less than 1, since then there are no encirclements of the critical point, -1, and hence closed-loop stability regardless of the phases involved. In theorem 4, the strategy is to restrict the arguments of the characteristic gain loci to lie between $-\pi$ and π, since then the loci never cross the negative real axis and closed-loop stability is inferred regardless of the moduli.

In theorem5, the characteristic gain loci are prevented from encircling the critical point by restricting their arguments in the frequency range $[0,\omega_b]$, and restricting their moduli in the frequency range $[\omega_b,\infty]$. The proofs of theorems 3 and 4 therefore, show that theorem 5 holds.

Remarks concerning theorem 3. The major steps of the proof of theorem 3 are as follows. It is assumed that the principal phases of $\Delta G(j\omega)$ form a convex cone, then from theorem 2 it follows that the eigenvalues of $\Delta G(j\omega)$ lie inside this cone. Similarly, it is assumed that $[I+G(j\omega)^{-1}]^{-1}$ has an associated convex cone within which its eigenvalues lie. It is then shown that the eigenvalues of $\Delta G(j\omega)[I+G(j\omega)^{-1}]^{-1}$, which are the characteristic gain loci of the perturbed closed-loop system, lie inside the "sum"[3] of the two convex cones modulo a phase modification, $\psi_m(\omega)$. In order to do this it is required that the sum of the convex cones be convex (condition (ii) of the theorem), and that the condition numbers satisfy condition (iii) of the theorem.

Note that condition (iii) implies that the condition number of the closed-loop frequency-response must be less than 2. In practice, for a design which aims at low interaction and accurate tracking through approximate diagonalization, $c_1(\omega)$ will be close to 1 allowing a large condition number for the perturbation. Note, however, that a large value of $c_2(\omega)$ corresponds to a large value of $\psi_m(\omega)$, which from condition (iv) imposes a greater restriction on $\varepsilon_1(\omega)$ as a phase lag in the perturbation. As $c_2(\omega)$ is reduced the phase lag in the perturbation can be increased; that is, condition number can be traded for phase and vice versa. Insofar as gain is concerned only the condition number is restricted, and so in theory the perturbation can have infinitely large gains.

---

[3] The sum of two convex cones is here defined as the cone formed from the products of all complex numbers in one cone with those in the other.

It should also be noted that $c_2(\omega)<\infty$, implies that $\Delta G(j\omega)$ is non-singular.

4.2 Concluding remark

The small phase theorem in conjunction with the small gain theorem can be used to identify a larger class of perturbations of the open-loop system for which the closed-loop system remains stable. The techniques are therefore an improvement on [8], [9] and [10], in which the robustness of the closed-loop stability property is characterized by singular values alone. It is also comforting to observe that for single-input, single-output systems these new measures of "robustness" are exactly the gain and phase margin measures used to characterize the closeness of the Nyquist diagram to the critical point, -1. If the techniques of [8], [9] and [10] are applied to scalar systems then the lack of phase information is an obvious disadvantage.

5. REFERENCES

[1] I.M.Horowitz, Synthesis of Feedback Systems. New York: Academic, 1963.

[2] J.B.Cruz,Jr. and W.R.Perkins,"A New Approach to the Sensitivity Problem in Multivariable Feedback Design",IEEE Trans. Automat. Contr.,vol.AC-9, pp.216-223,1964.

[3] W.R.Perkins and J.B.Cruz,Jr.,"Sensitivity Operators for Linear Time-Varying Systems", Sensitivity Methods in Control Theory. Edited by L.Radanović. Oxford: Pergamon, 1966.

[4] H.Nyquist,"Regeneration theory",Bell System Tech. J.,vol.11,pp.126-147, 1932.

[5] J.F.Barman and J Katzenelson,"A generalized Nyquist-type stability criterion for multivariable feedback systems",Int. J. Contr.,vol.20,no.4, pp.593-622,1974.

[6] A.G.J.MacFarlane and I.Postlethwaite,"The generalized Nyquist stability criterion and multivariable root loci",Int. J. Contr.,vol.25,no.1, pp.81-127,1977.

[7] C.A.Desoer and Y-T.Wang,"On the generalized Nyquist stability criterion", IEEE Trans. Automat. Contr.,vol.AC-25,no.2,pp.187-196,1980.

[8] J.C.Doyle,"Robustness of Multiloop Linear Feedback Systems",Proc. 17th CDC,San Diego, CA,1979.

[9] A.J.Laub,"An Inequality and Some Computations Related to the Robust Stability of Linear Dynamic Systems",IEEE Trans. Automat. Contr.,vol. AC-24,pp.318-320,1979.

[10] D.W.Nuzman and N.R.Sandell,Jr.,"An Inequality Arising in Robustness Analysis of Multivariable Systems",IEEE Trans. Automat. Contr.,vol. AC-24,pp.492-493,1979.

[11] I.W.Sandberg,"On the L_2-boundedness of solutions of nonlinear functional equations",Bell System Tech. J.,vol.43,pt.II,pp.1581-1599,1964.

[12] G.Zames,"On the Input-Output Stability of Time Varying Nonlinear Feedback Systems",pt.I,IEEE Trans. Automat. Contr.,vol.AC-11,no.2,pp.228-238,1966.

[13] C.A.Desoer and M.Vidyasagar, Feedback Systems:Input-Output Properties. New York: Academic, 1975.

[14] F.R.Gantmacher, Theory of Matrices,vol.1. New York: Chelsea, 1959.

[16] G.W.Stewart, Introduction to Matrix Computations. New York: Academic, 1973.

[17] I.Postlethwaite, J.M.Edmunds and A.G.J.MacFarlane,"Principal gains and principal phases in the analysis of linear multivariable feedback systems",Proc. 1980 JACC, San Francisco,CA; to appear in IEEE Trans. Automat. Contr.,vol.AC-26,no.1,Feb,1981.

[18] I.Postlethwaite and A.G.J.MacFarlane, A Complex Variable Approach to the Analysis of Linear Multivariable Feedback Systems. Berlin: Springer, 1979.

ON THE ROBUSTNESS OF OUTPUT FEEDBACK CONTROL METHODS TO MODELING ERRORS[1]

Hassan K. Khalil
Dept. of Electrical Engineering & Systems Science
Michigan State University
East Lansing, MI 48824

Abstract: Singular Perturbation analysis is employed to study the robustness of output feedback control schemes for linear time-invariant systems. The effect of designing the control strategy using a model that neglects unknown parasitic elements is considered. It is shown that such a strategy may in general be destabilizing even when the neglected modes are asymptotically stable. Special structures that will not permit this situation and will guarantee asymptotic stability are illustrated.

1. Introduction

A basic issue in the control of linear time-invariant systems is the investigation of the effect of modeling errors on the performance of the control system. Generally speaking, there are two models: the actual model and the design model. The actual model has the closest fidelity to the actual physical system, and usually encompasses unknown parameters. The design model is used by the designer to derive a control strategy, and represents a simplified version of the actual model where some of the modes may be neglected or some of the parameters are fixed at nominal values. It is of interest to study the effect of the discrepancy between the two models on the performance of the control system.

In this paper we consider typical modeling errors resulting from neglecting fast parasitic modes. A suitable approach to study the effect of this kind of errors is to employ singular perturbation analysis. Then, the actual model is represented as

---

[1] This work is supported by the National Science Foundation under Grant no. ENG-7912152.

$$\dot{x} = A_{11}x + A_{12}z + B_1u \tag{1a}$$

$$\mu\dot{z} = A_{21}x + A_{22}z + B_2u \tag{1b}$$

$$y = C_1x + C_2z \tag{1c}$$

where $x \in R^n$ is the dominant state vector, $z \in R^m$ is the state vector of the parasitic elements, $u \in R^r$ is the input vector, $y \in R^l$ is the output vector, and $\mu > o$ is a positive small scalar representing the parasitic elements (e.g., small inductances, capacitances, inertias, etc.). As usual in practice, the parasitic elements may be neglected. This is equivalent to setting $\mu = o$ in (1), and results in the reduced-order model

$$\dot{x} = A_ox + B_ou \tag{2a}$$

$$y = C_ox + D_ou \tag{2b}$$

where

$$A_o = A_{11} - A_{12}A_{22}^{-1}A_{21}, \qquad B_o = B_1 - A_{12}A_{22}^{-1}B_2$$

$$C_o = C_1 - C_2A_{22}^{-1}A_{21}, \qquad D_o = -C_2A_{22}^{-1}B_2$$

and where A_{22} is assumed to be nonsingular. Equation (2) is the design model. It is assumed that the designer knows (2) but does not know the complete description (1). If the reduced-order model (2) is used to design a control strategy, the robustness of that strategy can be investigated by applying it to the actual model (1) and performing asymptotic analysis of the performance as $\mu \rightarrow o$. If, for sufficiently small μ, the actual performance is arbitrarily close to the one predicted by the design model, it is reasonable to say that the control strategy is robust with respect to this kind of modeling errors.

It is a fact that the first requirement of any feedback control strategy is to stabilize the closed-loop system. In this paper we concentrate our attention on studying the stability of the actual system (1) when output feedback strategies, designed using the reduced-order model (2), are applied. To motivate our work let us briefly review the corresponding problem for state feedback strategies. It is well known (c.f.[1]) that a state feedback law

$$u = K_ox \tag{3}$$

designed to stabilize the reduced-order system (2), i.e., Re $\lambda(A_o + B_oK_o) < o$, will stabilize the actual system (1) for sufficiently small μ

if Re $\lambda(A_{22}) > o$, i.e., if the neglected fast modes are asymptotically stable. This simple result provides theoretical justification for neglecting the parasitic elements and agrees with the intuition that neglecting fast asymptotically stable modes will not cause a stability problem.

In this paper we show that a similar conclusion for output feedback control does not hold in general. We also identify special structures where it will hold.

2. Preliminaries

Consider the singularly perturbed system

$$\dot{x} = \Gamma_1 x + \Gamma_2 z \tag{4a}$$

$$\mu \dot{z} = \Gamma_3 x + \Gamma_4 z \tag{4b}$$

An eigenvalue property of this system is given by the following lemma which is recalled from [2].

Lemma: If Γ_4 is nonsingular, then the eigenvalues of (4), for sufficiently small μ, are approximated by

$$\lambda(\Gamma_o)\ (1 + O(\mu))$$

and

$$\frac{1}{\mu}\ \lambda(\Gamma_4)\ (1 + O(\mu))$$

where $\Gamma_o = \Gamma_1 - \Gamma_2 \Gamma_4^{-1} \Gamma_3$.

A direct consequence of this lemma is that if Γ_4 is nonsingular then necessary and sufficient conditions for the existence of $\mu^* > o$, such that (4) is asymptotically stable for $\mu \in (o, \mu^*)$, are given by

$$\text{Re } \lambda(\Gamma_4) < o \tag{5}$$

and

$$\text{Re } \lambda(\Gamma_o) > o. \tag{6}$$

Therefore, the procedure we will use in the next section to investigate the asymptotic stability of singularly perturbed systems of the form (4) comprises three steps. First, verify that Γ_4 is nonsingular; this is needed to make sure that Γ_o is well-defined. Second; verifty condition (5). Third, compute Γ_o and verify condition (6).

3. Output Feedback Control

Consider a physical system described by the actual model (1) where A_{22} is nonsingular. The reduced-order model (2), obtained by neglecting the parasitic elements, is known. Suppose that only the output vector y is measurable. A fairly general output feedback control scheme is given by

$$\dot{v} = G_1 v + G_2 y + G_3 u \tag{7}$$

$$u = F_1 v + F_2 y \tag{8}$$

where (7) is a dynamic compensator of order s; $o \le s \le n$. The extreme case s=o represents static gain feedback. Since the output equation (2b) has a direct feedthrough from the input u, we require that the matrix F_2 yields nonsingular $(I - F_2 D_o)$. This guarantees that u given by (8) will be uniquely defined. The closed-loop model known to designer is given by

$$\begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \mathbf{W}\begin{pmatrix} x \\ v \end{pmatrix} \triangleq \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{pmatrix}\begin{pmatrix} x \\ v \end{pmatrix}$$

where

$$W_{11} = A_o + B_o (I - F_2 D_o)^{-1} F_2 C_o$$

$$W_{12} = B_o (I - F_2 D_o)^{-1} F_1$$

$$W_{21} = [G_2 + (G_3 + G_2 D_o)(I - F_2 D_o)^{-1} F_2] C_o$$

$$W_{22} = G_1 + (G_3 + G_2 D_o)(I - F_2 D_o)^{-1} F_1$$

It is assumed that the choice of G_1, G_2, G_3, F_1 and F_2 will guarantee that

$$\mathrm{Re}\ \lambda(W) < o \tag{10}$$

otherwise the feedback control does not make any sense.

We now consider the application of the feedback control (7) and (8) to the actual system (1). The actual closed-loop system is given by the singularly perturbed equation (4) where (x,v) is the dominant state, z is the fast state, and

$$\Gamma_1 = \begin{pmatrix} (A_{11} + B_1 F_2 C_1) & B_1 F_1 \\ (G_2 + G_3 F_2) C_1 & (G_1 + G_3) \end{pmatrix},$$

$$\Gamma_2 = \begin{pmatrix} A_{12} + B_1 F_2 C_2 \\ (G_2 + G_3 F_2) C_2 \end{pmatrix},$$

$$\Gamma_3 = (A_{21} + B_2 F_2 C_1,\ B_2 F_1)$$

$$\Gamma_4 = A_{22} + B_2 F_2 C_2.$$

To investigate the asymptotic stability of this system we start by verifying that Γ_4 is nonsingular. This follows from the nonsingularily of $(I - F_2 D_o)$. In fact, Γ_4^{-1} is given by

$$\Gamma_4^{-1} = A_{22}^{-1} \left[I - B_2 (I - F_2 D_o)^{-1} F_2 C_2 A_{22}^{-1} \right].$$

Moreover, it can be shown, after lengthy manipulation, that

$$\Gamma_o \overset{\Delta}{=} \Gamma_1 - \Gamma_2 \Gamma_4^{-1} \Gamma_3 = \mathbf{W}. \tag{11}$$

From (10) and (11) we find that condition (6) is satisfied. Thus the only remaining task is to verify condition (5). Here where the trouble starts. What we need is

$$\operatorname{Re} \lambda (A_{22} + B_2 F_2 C_2) < o \tag{12}$$

but there is no guarantee that (12) will be satisfied. In other words, a designer who chooses F_2, among other parameters, to stabilize the reduced-order closed-loop model (9) can not be sure that (12) is satisfied even when $\operatorname{Re} \lambda (A_{22}) < o$. This will be demonstrated by simple examples later. If A_{22}, B_2 and C_2 are known to the designer he can restrict the class of permissible matrices F_2 to those for which (12) holds. But, as it is the case in many situations, A_{22}, B_2, and C_2 may not be known; and a general conclusion similar to the case of state feedback is not possible. In other words, neglecting fast asymptotically stable modes may cause instability in the case of output feedback.

Our next task is to look for special cases where this instability problem is not permitted. Let $\operatorname{Re} \lambda(A_{22}) < o$ and either $B_2 = o$ or $C_2 = o$, i.e., asymptotically stable fast modes that are either not directly fed from the input or not directly sensed at the output. In this case

(12) is satisfied. Another special case is given by the following proposition:

Proposition: If Re $\lambda(A_{22}) < o$, then the asymptotic stability of the class of output feedback control schemes (7) and (8) which do not have a static feedback from the output to the input, i.e., $F_2=o$, is robust with respect to neglecting the parasitic elements. The proposition is straight forward but it identifies a class of output feedback methods that enjoys the same robustness property as state feedback.

To illustrate the previous discussion and give numerical examples, we consider the following standard output feedback schemes.

1. Static gain feedback

$$u = K_o\, y \tag{13}$$

This is a special case of (8) with $F_1=o$ and $F_2=K_o$. Obviously, in this case one may expect instability problem as shown by the following example.

Example 1:

Consider the second-order system

$$\begin{aligned} \dot{x} &= x + z + u \\ \mu\dot{z} &= x - z + u \\ y &= x + z \end{aligned} \tag{14}$$

for which Re $\lambda(A_{22}) < o$. Upon neglecting the parasitic element one obtains the first order model.

$$\begin{aligned} \dot{x} &= 2x + 2u \\ y &= 2x + u \end{aligned} \tag{15}$$

Using this first-order model as a design model, a designer has chosen the feedback control

$$u = 2\, y \tag{16}$$

which results in the asymptotically stable first-order system

$$\dot{x} = -6x \tag{17}$$

When the feedback control (16) is applied to the actual system (14), the closed-loop equations are

$$\begin{aligned} \dot{x} &= 3x + 3z \\ \mu\dot{z} &= 3x + z \end{aligned} \tag{18}$$

which is unstable as $\mu \to o$ because $\Gamma_4=1 > o$. In fact, in this simple example one may approximate the eigenvalues as $\mu \to o$ by -6 and $\frac{1}{\mu}$.

2. Output feedback using reduced-order (Luenberger) observer

An output feedback of the design model (2) using Luenberger observer is given by [3]

$$\dot{v} = Tv + H(y - D_o u) + Mu \tag{19a}$$

$$\hat{x} = Q_1(y - D_o u) + Q_2 v \tag{19b}$$

$$u = F_o \hat{x} \tag{19c}$$

where T and H are arbitrary, V is the solution of

$$VA_o - HC_o - TV = o,$$

$$[Q_1, Q_2] = \begin{bmatrix} C_o \\ V \end{bmatrix}^{-1},$$

and

$$M = VB_o$$

Comparing (19) with (7) and (8) we find that

$$F_2 = (I + F_o Q_1 D_o)^{-1} F_o Q_1 \neq o.$$

Hence, instability problem is possible as shown by the following example.

Example 2: Consider the third-order system

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} z + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u$$

$$\mu \dot{z} = x_1 - 4z + u \tag{20}$$

$$y = x_1 + z$$

The corresponding design model is

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1.25 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1.25 \end{pmatrix} u \tag{21}$$

$$y = 1.25\ x_1 + 0.25\ u.$$

Using the reduced-order observer design given in [4], a designer has derived the output feedback control

$$\dot{v} = -1.5v - 2(y - 0.25u) + 1.25u$$

$$\hat{x}_1 = 0.8(y - 0.25u) \qquad (22)$$

$$\hat{x}_2 = v + 2(y - 0.25u)$$

$$u = -2\,\hat{x}_1 - 2\,\hat{x}_2$$

which locates the observer pole at $-\frac{3}{2}$ and the regulator poles at $-\frac{3}{4} \pm j\,\frac{1}{4}\sqrt{11}$. When the feedback control (22) is applied to the actual system (20), it results in the closed-loop equations

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{v} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 15 & 1 & 5 \\ 22.5 & 0 & 7.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ 15 \\ 22.55 \end{pmatrix} z$$

$$\mu\dot{z} = (15,\ 0,\ 5) \begin{pmatrix} x_1 \\ x_2 \\ v \end{pmatrix} + 10\,Z$$

which is unstable for sufficiently small μ because $\Gamma_4 = 10 > 0$.

3. Output feedback using full-order observer

An output feedback of the design model (2) using a full-order observer is given by [4]

$$\dot{\hat{x}} = A_o\,\hat{x} + B_o\,u + K_o\,[y - D_o u - C_o\,\hat{x}] \qquad (23a)$$

$$u = F_o\hat{x} \qquad (23b)$$

Comparing (23) with (7) and (8) it is seen that $F_2 = o$. Hence, output feedback using full-order observers belongs to the class of robust output feedback schemes characterized in the proposition. It should be mentioned that the robustness of output feedback using full-order observers has been previously shown in [5, 6, 7].

References

1. J.H. Chow and P.V. Kokotovic, "A decompoistion of near-optimum regulators for systems with slow and fast modes," IEEE Trans. Automatic Control, Vol. AC-21, pp. 701-705, 1976.

2. J.H. Chow and P.V. Kokotovic, "Eigenvalue placement in two-time-scale systems," Large scale systems theory and applications, edited by G. guardabassi, A. Locatelli, ISA Public. (IFAC Symposium), 1976.

3. D.G. Luenberger, "Observing the states of a linear system," IEEE Trans. Mil. Electronics, Vol. MIL-8, pp. 74-80, 1964.

4. H. Kwakernaak and R. Sivan, "Linear Optimal Control Systems," WILEY-INTERSCIENCE, 1972.

5. B. Porter, "Singular perturbation methods in the design of observers and stabilizing feedback controller for multivariable linear systems," Electron. Lett., Vol. 10, Nov. 14, 1974.

6. M.J. Balas, "Observer stabilization of singularly perturbed systems," J. Guidance and Contr., Vol. 1, Jan. - Feb. 1978.

7. S.H. Javid, "Observing the slow states of a singularly perturbed system," IEEE Trans. Automatic Control, Vol. AC-25, pp. 277-280, 1980.

CONTROLLER DESIGN FOR PARAMETER SENSITIVITY REDUCTION IN LINEAR REGULATORS

RAMAKRISHNA YEDAVALLI and ROBERT E. SKELTON
School of Aeronautics and Astronautics
Purdue University
West Lafayette, IN 47907

ABSTRACT

This paper addresses the problem of parameter sensitivity reduction in linear regulators. It is shown that the minimization of a modified performance index that includes a quadratic term of trajectory sensitivity (state or output or control) subject to the augmented sensitivity system dynamics causes reduction not only in trajectory sensitivity(state or output or control) but also in *cost sensitivity*. By including quadratic terms of both output (state) sensitivity and control sensitivity, the approaches of trajectory sensitivity reduction and cost sensitivity reduction are united into one framework. A control design method, based on the linear optimal control formulation, is then presented and improvements of the proposed method are established.

1.0 INTRODUCTION

After the early and pioneering work on parameter sensitivity in linear multivariable control systems [1]-[5], considerable research has been focused on parameter sensitivity reduction in linear regulators. In this field, two methods are currently popular: i) the 'trajectory sensitivity' reduction, and ii) the 'cost (performance index) sensitivity' reduction. In both of these approaches the object is to achieve a trade off between optimality in the nominal performance (trajectory for the former and performance cost for the latter) and sensitivity to small parameter variations. With these two schools of thought, design methods for output (state) sensitivity reduction [6]-[11] and cost sensitivity reduction [12]-[13] are reported separately. There seems to be considerable lack of attention towards the problem of control sensitivity reduction. This paper combines these three aspects into one framework, presenting a control design method based on a new performance index that achieves the said unification. Advantages of the proposed method are substantiated.

2.0 SENSITIVITY SYSTEM DESCRIPTION

Consider the linear time invariant system

$$\dot{x}(t,\bar{p}) = A(\bar{p})x(t,\bar{p}) + B(\bar{p})u(t,\bar{p}) \ , \ x(o) = x_o \qquad (2.1)$$

$$y(t,\bar{p}) = C(\bar{p})x(t,\bar{p}) \qquad (2.2)$$

where the state vector is $x \varepsilon R^n$, the control is $u \varepsilon R^m$, the outputs $y \varepsilon R^k$ are the variables we wish to control. The system matrices A, B, C are assumed to be continuous functions of constant but uncertain parameters $p_1, p_2, \ldots, p_r$ with nominal value $p_i = \bar{p}_i$ $(i=1,2,\ldots,r)$. Let the vector of uncertain parameters be denoted by p where

$$p^T = [p_1 \ p_2 \ \ldots \ p_r] \qquad (2.3)$$

The initial condition vector x_o is assumed to be independent of the uncertain parameters. The matrix triple $(A(\bar{p}),B(\bar{p}),C(\bar{p}))$ is assumed to be completely controllable and observable.

Let the performance index for the system be given by

$$V(\bar{p}) = \int_0^\infty [y^T(t,\bar{p})Qy(t,\bar{p}) + u^T(t,\bar{p})Ru(t,\bar{p})]dt \tag{2.4}$$

where Q,R are (kxk) and (mxm) symmetric positive definite matrices respectively. We refer the system (2.1)-(2.4) as the 'Nominal system'. Henceforth the explicit time and parameter dependence of x,u,y,V and other related quantities will not be shown (except when deemed necessary).

Let $x_{p_i}, y_{p_i}, u_{p_i}$ and V_{p_i} denote the 'state trajectory sensitivity', 'output trajectory sensitivity', 'control trajectory sensitivity' and 'cost sensitivity' respectively with respect to p_i defined by $[\cdot]_{p_i} = \frac{\partial[\cdot]}{\partial p_i}$ evaluated at the niminal value $p_i = \bar{p}_i$. The 'state trajectory sensitivity vector', x_p is denoted by the nr-vector

$$(x_p)^T = [x_{p_1}^T \; x_{p_2}^T \; \ldots \; x_{p_r}^T] \quad , \quad x_p \varepsilon R^{nr} \tag{2.5}$$

The 'output trajectory sensitivity vector, y_p' and the 'control sensitivity vector, u_p' are similarly defined,

$$\text{i.e.}((\cdot)_p)^T \triangleq [(\cdot)_{p_1}^T \; (\cdot)_{p_2}^T \; \ldots \; (\cdot)_{p_r}^T] \, , \; y_p \varepsilon R^{kr} \; , \; u_p \varepsilon R^{mr} \tag{2.6}$$

The 'cost sensitivity vector, V_p' is an r vector given by

$$(V_p)^T = [V_{p_1} \; V_{p_2} \; \ldots \; V_{p_r}] \tag{2.7}$$

Defining, for an $\alpha x \beta$ matrix F,

$$\tilde{F} \triangleq \text{Block diag } [\ldots. \; F \; \ldots.] \text{ (r times)}$$

and noting that $[FH]_p = F_p H + \tilde{F} H_p$ (2.8)

the "closed-loop sensitivity system" can be described as

$$\dot{x}_p = A_p x + \tilde{A} x_p + B_p u + \tilde{B} u_p \quad , \quad x_p(o) = 0 \tag{2.9}$$

$$y_p = C_p x + \tilde{C} x_p \tag{2.10}$$

where the words "closed-loop" imply that u is a function of p. The 'composite system', obtained by augmenting the 'sensitivity system' (2.9)-(2.10) with the 'nominal system' (2.1)-(2.2) becomes

$$\dot{\mathcal{X}} = \mathcal{A}\mathcal{X} + \mathcal{B}_1 u + \mathcal{B}_2 u_p \quad , \quad \mathcal{X}(o) = \begin{bmatrix} x_o \\ o \end{bmatrix} \tag{2.11}$$

$$\mathcal{Y} = \mathcal{C}\mathcal{X}$$

where

$$\mathcal{X}^T = [x^T (x_p)^T] \, , \; \mathcal{Y}^T = [y^T (y_p)^T] \tag{2.12}$$

$$\mathcal{A} = \begin{bmatrix} A & 0 \\ A_p & \tilde{A} \end{bmatrix} \; , \; \mathcal{B}_1 = \begin{bmatrix} B \\ B_p \end{bmatrix} \; , \; \mathcal{B}_2 = \begin{bmatrix} 0 \\ \tilde{B} \end{bmatrix} \text{ and } \mathcal{C} = \begin{bmatrix} C & o \\ C_p & \tilde{C} \end{bmatrix} \tag{2.13}$$

In the case of 'open loop sensitivity system', where the control u is considered to be independent of the parameter vector p, the composite system is the same as above except that

$$\dot{x} = Ax + B_1 u \tag{2.14}$$

It may be noted that the matrix pair (A,C) is always completely observable [14] and the matrix pair (A,B_1) is uncontrollable for sufficiently large number of uncertain parameters [15].

3.0 PERFORMANCE INDEX FOR SENSITIVITY REDUCTION

A common method to reduce output (state) trajectory sensitivity is to include a quadratic term of the output (state) trajectory sensitivity in the cost functional V of (2.4) i.e.

$$V_{st} = \int_0^\infty (y^T Q y + (y_p)^T Q_s y_p + u^T R u)dt \tag{3.1}$$

and to design a controller that minimizes V_{st}. For cost sensitivity reduction, an index of the form

$$V_c = V + \sum_i \alpha_i V_{p_i}^2 \tag{3.2}$$

is often specified and a controller is designed to minimize this index.

In most of these methods reported [6]-[11], the relationship between output (state) trajectory sensitivity and cost sensitivity has not been formally established and exploited. Also no attention has been paid to the equally important problem of control sensitivity reduction. In this section, a new performance index is specified which, while retaining the structure of the integral quadratic cost functional, reduces output (state) sensitivity, control sensitivity and cost sensitivity together, thereby unifying the three approaches into one framework. One major contribution of this paper is the following theorem.

Theorem 1: *The performance index*

$$V_s = \int_0^\infty (y^T Q y + (y_p)^T Q_s y_p + (u_p)^T R_s u_p + u^T R u)dt \tag{3.3}$$

is an upper bound to the L_2 *norms* $\|y_p\|^2_{Q_s}$, $\|u_p\|^2_{R_s}$ *and to the absolute norm* $\sum_{i=1}^{r} |\alpha_i V_{p_i}|$ *where* Q_s = block diag $[\ldots.\ Q_{s_{ii}}\ \ldots.]$, R_s = block diag $[\ldots.\ R_{s_{ii}}\ \ldots.]$ $Q_{s_{ii}}, R_{s_{ii}}$ *are positive definite matrices, and* α_i *are nonzero scalars such that the matrix*

$$S_c = \begin{bmatrix} S & \pm S_{c_{12}} \\ \pm S_{c_{12}}^T & \bar{S} \end{bmatrix} \tag{3.4a}$$

is positive definite, where

$$\bar{S} = \text{block diag}[\ldots.\ \bar{S}_{ii}\ \ldots.]\ ,\ \bar{S}_{ii} = \begin{bmatrix} Q_{s_{ii}} & 0 \\ 0 & R_{s_{ii}} \end{bmatrix}$$

$$S_{c_{12}} = [\alpha_1 S\ \alpha_2 S\ \ldots\ \alpha_r S]\ ,\ S = \text{block diag}[Q,R]. \tag{3.4b}$$

Corollary 1: The performance index

$$V_{st} = \int_0^\infty (y^TQy + (y_p)^TQ_sy_p + u^TRu)dt \tag{3.5}$$

is an upper bound to the L_2 norm $\| y_p\|^2_{Q_s}$ and to the absolute norm $\sum_{i=1}^{r} |\alpha_i V_{p_i}|$ where α_i has the property that $V_{st} \pm \alpha_i V_{p_i} > 0$. In the present case, $\bar{S}_{ii} = \begin{bmatrix} Q_{s_{ii}} & o \\ o & o \end{bmatrix}$.

Corollary 2: The performance index

$$V_{su} = \int_0^\infty (y^TQy + (u_p)^TR_su_p + u^TRu)dt \tag{3.6}$$

is an upper bound to the L_2 norm $\| u_p\|^2_{R_s}$ and to the absolute norm $\sum_{i=1}^{r} |\alpha_i V_{p_i}|$ where α_i has the same property as in Corollary 1, except that, in the present case $\bar{S}_{ii} = \begin{bmatrix} o & o \\ o & R_{s_{ii}} \end{bmatrix}$.

Proofs of Theorem 1 and Corollaries 1 and 2 are provided in [14].

In the special case

$$\begin{aligned} Q_s &= \text{diag}[\rho_1 Q,\ldots,\rho_r Q] \\ R_s &= \text{diag}[\rho_1 R,\ldots,\rho_r R] \end{aligned} \tag{3.7}$$

the scalars α_i in (3.4) satisfy [14]

$$\sum_{i=1}^{r} \alpha_i^2/\rho_i < 1 \tag{3.8}$$

If the parameter p_i has the variance

$$E(p_i - \bar{p}_i)^2 = \sigma^2_{p_i}$$

then a reasonable choice for the weighting ρ_i in (3.7) is $\sigma^2_{p_i}$. With this choice (3.3) reduces to the nominal (2.4) when parameters are certain (σ_i=0).

Consider the Taylor's series expansion of the cost function (2.4) about $\bar{p}$.

$$V(p) = V(\bar{p}) + \Delta p^TV_p + \frac{1}{2}\,\Delta p^TV_{pp}\Delta p + \ldots \tag{3.9}$$

where $\Delta p \triangleq p-\bar{p}$ and V_{pp} is the Hessian whose ij element is

$$[V_{pp}]_{ij} = \left.\frac{\partial^2 V}{\partial p_i \partial p_j}\right|_{p=\bar{p}} , \quad i,j=1,\ldots r$$

To gain insight into the relationship between the *trajectory* sensitivity terms $\| y_p\|^2_{Q_s}$, $\| u_p\|^2_{R_s}$ in (3.3) and the sensitivity of the *cost* function

(2.4) we observe the following.

Theorem 2: *If the parameter variations are such that second order sensitivities* $\frac{\partial^2 y}{\partial p_i^2}$, $\frac{\partial^2 u}{\partial p_i^2}$ *are zero, then*

$$\int_0^\infty (\| y_p \|^2_{Q_s} + \| u_p \|^2_{R_s})dt = \frac{1}{2} \sum_{i=1}^{r} \rho_i \frac{\partial^2 V}{\partial p_i^2} \tag{3.10}$$

where V *refers to* (2.4) *and* Q_s, R_s *are defined by* (3.7). For proof see [14].

Minimization of the performance index V_s serves the purpose of achieving a trade-off between optimality of V at the nominal value and insensitivity in the sense of the norms $\| y_p \|^2_{Q_s}$, $\| u_p \|^2_{R_s}$ and $\sum_{i=1}^{r} |\alpha_i V_{p_i}|$. It is noted from Theorem 1 and the two corollaries that the inclusion of either input or output (state) trajectory sensitivity in the integrand of the original cost functional, facilitates cost sensitivity reduction in addition to the intended trajectory sensitivity reduction. Even though the performance index (3.5) has been in use in the context of trajectory sensitivity reduction for some time, the inter relationships between these different sensitivities, as delineated above, has not been formally established and exploited. The performance index V_s also has the advantage that it retains the form of an integral quadratic cost functional rendering itself amenable to the standard quadratic optimization theory. The approaches of trajectory sensitivity reduction, and cost sensitivity reduction are thus garnered into one framework. It is to be noted that cost sensitivity reduction is obtained in this formulation as a 'by-product' of trajectory sensitivity reduction, although the norm (3.4) or (3.8) is somewhat more restrictive. Relation (3.8) suggests that higher the parameter variance, higher is the weighting on cost sensitivity.

4.0 CONTROLLER DESIGN

The performance index V_s of (3.3) can be expressed in terms of state, control and their sensitivities as

$$V_s = \int_0^\infty [x^T (u_p)^T u^T] \begin{bmatrix} U_s & o & o \\ o & R_s & o \\ o & o & R \end{bmatrix} \begin{bmatrix} x \\ u_p \\ u \end{bmatrix} dt \tag{4.1}$$

where $U_s = \begin{bmatrix} U_{11} & U_{12} \\ U_{12}^T & U_{22} \end{bmatrix}$, $U_{11} = C^T QC + (C_p)^T Q_s C_p$

$$U_{12} = (C_p)^T Q_s \tilde{C} \text{ and } U_{22} = \tilde{C}^T Q_s \tilde{C} \tag{4.2}$$

An attempt to use the standard linear optimal regulator theory to this modified index, prompts one to consider a control law of the form

$$u = G_1 x + G_2 x_p \tag{4.3}$$

When the control u takes the form given in (4.3), it is not possible to introduce into the formulation the exact trajectory sensitivity x_p by use of a finite member of differential equations [6]. Assuming that the trajectory sensitivities of order two (and higher) are zero one can get a finite dimensional model for generating the trajectory sensitivity vector, although this is not always justified [6]. Even under this assumption Newmann [7] noted that the design problem posed in (4.1)-(4.3) is no longer in the standard linear regulator format owing to the dependence of the plant matrix on the control gain matrix. Newmann [7] then proved that no finite state independent gains G_1 and G_2 exist which satisfy the necessary conditions for an optimum.

The proposed approach in this paper is to cast an approximate problem in a finite dimensional setting. This is done by approximating the control sensitivity vector u_p as follows:

Approximation A: *We substitute the closed loop sensitivity of the standard optimal linear regulator control in the place of* u_p *of* (4.3). *i.e. we use*

$$u_{pa} = \tilde{G}\, x_{pa} \tag{4.4}$$

where

$$u = Gx\ ,\ G = -R^{-1}B^TK \qquad KA + A^TK - KBR^{-1}B^TK + C^TQC = 0 \tag{4.5}$$

is *the standard optimal control solution without sensitivity considerations.*

Employing the suggested approximation, the trajectory sensitivity equation becomes

$$\dot{x}_{pa} = A_p x + (\tilde{A}+\tilde{B}\tilde{G})x_{pa} + B_p u \tag{4.6}$$

where x_{pa} denotes the approximate nature of the trajectory sensitivity term. The modified problem formulation is then:

minimize
w.r.t.u

$$V_s = \int_0^\infty (x_a^T U_s x_a + (u_{pa})^T R_s (u_{pa}) + u^T Ru)dt$$

$$= \int_0^\infty (x_a^T U_c x_a + u^T Ru)dt \tag{4.7}$$

subject to

$$\dot{x}_a = A_c x_a + B_c u\ ,\ x_a(o) = \begin{bmatrix} x_o \\ o \end{bmatrix} \tag{4.8a}$$

$$u_{p_a} = G_a x_a \tag{4.8b}$$

where

$$x_a^T = [x^T (x_{pa})^T);\ A_c = \begin{bmatrix} A & \tilde{0} \\ A_p & \tilde{A}+\tilde{B}\tilde{G} \end{bmatrix};\ B_c = \begin{bmatrix} B \\ B_p \end{bmatrix};\ G_a = [0\ \tilde{G}] \tag{4.8c}$$

$$U_c = \begin{bmatrix} U_{11} & U_{12} \\ U_{12} & U_{22} \end{bmatrix} \qquad U_{22} = \tilde{C}^T Q_s \tilde{C} + \tilde{G}^T R_s \tilde{G}$$

and subscript 'c' reflects the closed loop version of the sensitivity system description.

Theorem 3: *The matrix pair* (A_c, B_c) *is always stabilizable.*

The proof is provided in [14].

Since the pair (A_c, B_c) is stabilizable and (A_c, C) is observable (recall from Section 2), the solution to the optimal control problem in (4.7)-(4.8) is

$$u = G_c x_a = G_{c_1} x + G_{c_2} x_{pa} \tag{4.9a}$$

$$G_c = -R^{-1} B_c^T K_c \tag{4.9b}$$

$$K_c A_c + A_c^T K_c - K_c B_c R^{-1} B_c^T K_c + U_c = 0 \tag{4.9c}$$

We are now prepared to summarize the design procedure.

The Sensitivity Regulator Algorithm (SRA)

Step 1: Compute the optimal control gain G given by the equations (4.5).

Step 2: Form the matrices A_c, B_c and U_c as given in (4.8c).

Step 3: The final control is given by (4.9).

An advantage of SRA design over the open loop sensitivity system design [16] is that it does not assume the open loop system matrix A to be stable in order to guarantee solution.

The closed loop system with the SRA control is given by

$$\dot{x}_a = A_s x_a \qquad x_a(o) = \begin{bmatrix} x_o \\ o \end{bmatrix} \tag{4.10}$$

where the matrix

$$A_s = \begin{bmatrix} A+BG_{c_1} & BG_{c_2} \\ A_p+B_pG_{c_1} & \tilde{A}+\tilde{B}\tilde{G}+B_pG_{c_2} \end{bmatrix} \tag{4.11}$$

is asymptotically stable.

Once the control loop is closed, we can find the exact closed loop trajectory sensitivity by differentiating equation (4.10) with respect to p. Accordingly, we get

$$\begin{bmatrix} \dot{x}_p \\ (\dot{x}_{pa})_p \end{bmatrix} = \begin{bmatrix} A_p+B_pG_{c_1} & B_pG_{c_2} \\ o & o \end{bmatrix} \begin{bmatrix} x \\ x_{pa} \end{bmatrix} + \begin{bmatrix} (A+BG_{c_1})^{\sim} & (BG_{c_2})^{\sim} \\ (A_p+B_pG_{c_1})^{\sim} & \tilde{A}_c \end{bmatrix} \begin{bmatrix} x_p \\ (x_{pa})_p \end{bmatrix} ; \begin{bmatrix} x_p(o) \\ (x_{pa})_p(o) \end{bmatrix} = \begin{bmatrix} o \\ o \end{bmatrix} \tag{4.12}$$

where $A_c = \tilde{A}+\tilde{B}\tilde{G}+B_pG_{c_2}$

The difference between x_p and x_{pa} is shown by Theorem 4 to converge to zero.

Theorem 4: *The error between the exact closed loop trajectory sensitivity* x_p *and the modeled (approximate) closed loop trajectory sensitivity* x_{pa} *is asymptotically convergent to zero.*

Proof: Since the matrix A_s of (4.11) is asymptotically stable, from equations (4.10) and (4.12) we observe that both x_p and x_{pa} are stable trajectories. Hence

$$\tilde{x}_p \triangleq x_p - x_{pa} \tag{4.13}$$

is asymptotically convergent to zero.

4.1 Illustration of SRA

Example 1: The following table summarizes the results of the application of SRA and Byrne and Burke method [10] to the first order system given by

$$\dot{x} = \frac{1}{\omega^2} x + \frac{1}{\omega} u \quad , \quad x_o = 1$$

where 'ω' is the uncertain parameter with $\bar{\omega} = 1$ and the performance index is given by

$$V = \int_o^\infty (x^2 + u^2)dt \ , \ V_s = \int_o^\infty (x^2 + x_\omega^2 + u_\omega^2 + u^2)dt$$

TABLE 1

Quantity	Standard Linear Regulator $Q_s=0$, $R_s=0$	SRA $Q_s=1$, $R_s=1$	SRA $Q_s=1$, $R_s=0$	Byrne & Burke's Method $Q_s=1$, $R_s=0$
V	2.414	2.439	2.416	2.417
$\int_o^\infty x_\omega^2 \, dt$	0.015*	0.004	0.011	0.016*
$\int_o^\infty u_\omega^2 \, dt$	0.088$^+$	0.018	0.063	0.097$^+$
$\lvert V_\omega \rvert$	0.707‡	0.374	0.622	0.780‡
$\int_o^\infty u^2 \, dt$	2.061	2.016	2.045	2.078

Compare the * values to see that trajectory sensitivity is not reduced by Byrne and Burke's method. Comparisons of the + and † values also indicate that control and cost sensitivity also increased, whereas the SRA decreased all three sensitivities.

Example 2: We now consider the same example reported in [8]-[10].

$$\dot{x} = a^2 x + u \quad , \quad x_o = 1 \quad , \quad \bar{a} = 1$$

$$V = \frac{1}{2} \int_o^\infty (2x^2 + 2x^2) \, dt \quad , \quad V_{st} = \frac{1}{2} \int_o^\infty (2x^2 + 2x_a^2 + 2u^2) dt \ , \ S = \int_o^\infty x_a^2 \, dt$$

The results are given in TABLE 2.

TABLE 2

	Standard Linear Regulator	Kriendler	Rao and Soudack	Fleming and Newmann	Byrne and Burke	Yedavalli and Skelton
V	2.414	2.450	2.443	2.478	2.432	2.466
S	0.354	0.176	0.189	0.140	0.216	0.154

Inspection of Table 2 indicates that for this example, the proposed method is the second best in reducing trajectory sensitivity. However, it is important to realize that the Fleming and Newmann method requires extensive computation (the method increases the order of the optimization problem to 4n and requires solution of many Liapunov equations) whereas the SRA method is somewhat simpler, both computationally and conceptually, compared to all methods presented in Table 2.

5.0 CONCLUSIONS

A unified approach to parameter sensitivity-reduction in linear regulator problems is presented, which simultaneously reduces output (state) trajectory sensitivity, control trajectory sensitivity, and cost sensitivity. A simple control design algorithm, labeled Sensitivity Regulator Algorithm (SRA), is presented and the improvement of the proposed method over other existing methods is established.

The proposed algorithm is specialized to deterministic systems assuming the sensitivity functions are generated for feedback purposes. Extension of the algorithm to linear stochastic regulators (the separation principle does not hold in the presence of parameter uncertainties), is given in [14].

REFERENCES

1. Cruz, J. B. and Perkins, W. R., "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design", IEEE Trans. Autom. Control, Vol. AC-9, 1964, pp. 216-223.

2. Perkins, W. R. and Cruz, J. B., "The Parameter Variation Problem in State Feedback Control Systems", Trans. ASME, Vol. 87, 1965, pp. 120-124.

3. Kokotovic, P., Heller, J., Sannuti, P., "Sensitivity Comparison of Optimal Controls", International Journal of Control, 1969, pp. 111-117.

4. Dorato, P., "On Sensitivity in Optimal Control Systems", IEEE Trans. Autom. Control, Vol. AC-8, 1963, pp. 256-257.

5. Kreindler, E., "On the Definition and Application of the Sensitivity Function", J. of Franklin Inst., Vol. 285, 1968, pp. 26-36.

6. Kreindler, E., "Formulation of the Minimum Trajectory Sensitivity Problem", IEEE Trans. Autom. Control, Vol. AC-14, April 1969, pp. 206-207.

7. Newmann, M. M., "On Attempts to Reduce the Sensitivity of the Optimal Regulator to a Parameter Change", International Journal of Control, Vol. 11, No. 6, 1970, pp. 1079-1084.

8. Rao, S. G. and Soudack, A. C., "Synthesis of Optimal Control System with Near Sensitivity Feedback", IEEE Trans. Autom. Control, Vol. AC-16, April 1971, pp. 194-196.

9. Fleming, P. J. and Newmann, M. M., "Trajectory Sensitivity Reduction in the Optimal Linear Regulator", in Recent Mathematical Developments in Control, D. J. Bell, Ed., New York, Academic, 1973, pp. 137-151.

10. Byrne, P. C. and Burke, M., "Optimization with Trajectory Sensitivity Considerations", IEEE Trans. Autom. Control, Vol. AC-21, April 1976, pp. 282-283.

11. Subbayyan, R. et al., "An Approach for Sensitivity Reduced Design of Linear Regulators", International Journal of Systems Sciences, Vol. 9, No. 1, 1978, pp. 65-75.

12. Yahagi, T., "Optimal Output Feedback Control with Reduced Performance Index Sensitivity", International Journal of Control, Vol. 25, No. 5, 1977, pp. 769-783.

13. Missaghie, M. M. and Fairman, F. W., "Sensitivity Reducing Observers for Optimal Feedback Control", IEEE Trans. Autom. Control, Vol. AC-22, No. 6, December 1977.

14. Yedavalli, Ramakrishna, "Control Design for Parameter Sensitivity Reduction in Linear Regulators: Application to Large Space Structures", Ph.D. Thesis, School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN, 1980, Advisor, R. E. Skelton.

15. Guardabassi, G., Locatelli, A., and Rinaldi, S., "Structural Properties of Sensitivity Systems", Journal of Franklin Institute, Vol. 294, No. 4, October 1972, pp. 241-248.

16. Sesak, J. R., "Sensitivity Constrained Linear Optimal Control, Analysis and Synthesis", Ph.D. Thesis, University of Wisconsin, December 1974.

ON SENSITIVITY OF NONLINEAR DIGITAL SYSTEMS

MANSOUR ESLAMI
Department of Electrical Engineering
State University of New York at Stony Brook
Stony Brook, New York 11794

ABSTRACT

In this paper first several upper bounds for parameter variations of a set of linear constant coefficients difference equations, encountered in digital systems, for which the system remains stable are developed. These sensitivity inequalities can then be used to estimate the largest variations of the coefficients of a convergent polynomial, pertinent to the characteristic equation of a digital filter, for which the filter remains stable. A computer program for numerical calculation of these upper bounds is proposed. Extensions of these results to the nonlinear difference equations are suggested subsequently.

1. INTRODUCTION

From a practical point of view it is absolutely essential to estimate the largest parameter variations for which the system remains stable. This information gives every possible operating condition associated to various sets of system parameters for which the system remains stable. Correspondingly, in this paper several new upper bounds for parameter variations are developed. The extensions of these results to the nonlinear discrete-time system are developed next. The continuous-time version of these results is presented in [3].

The organization of this paper is as follows. In Section 2, the new results are developed. In Section 3, the computational considerations related to the numerical evaluation of these results are discussed. In Section 4, the results of this paper are extended to the determination of the necessary condition of the absolute stability of a set of nonlinear difference equations. The conclusions are summarized in Section 5.

2. NEW UPPER BOUNDS

The analysis in this paper stems from the Direct Method of Lyapunov [6], for which an exhaustive survey paper has just been published [2]. Thus, before proceeding to our new results this technique is reviewed.

THEOREM 1, [7]. All the eigenvalues of the matrix A lie in the unit circle if and only if for an arbitrary symmetric positive-definite matrix Q, the unique solution P of the matrix equation

$$A^TPA + Q = P, \tag{1}$$

is symmetric and positive definite.

This matrix A is called a convergent matrix. Our new results are as follows.

THEOREM 2. For a perturbation ΔA of a convergent matrix A of (1), (A + ΔA) remains a convergent matrix if

$$\Delta A^T(P + 2PAQ^{-1}A^TP)\Delta A < Q/2, \tag{2}$$

for an arbitrary symmetric positive-definite matrix Q, and the corresponding P as given by (1).

PROOF. Equation (1) can be exactly rewritten as

$$(A + \Delta A)^TP(A + \Delta A) - P = -Q + \Delta A^TP\Delta A + \Delta A^TPA + A^TP\Delta A. \tag{3}$$

Citing the Lyapunov's theorem again it is seen that (A + ΔA) remains convergent if

$$\tilde{Q} \triangleq Q - \Delta A^TP\Delta A - \Delta A^TPA - A^TP\Delta A > 0. \tag{4}$$

To put (4), which is stronger than (2) and often as useful as (2), in the desired form of (2), we use the following inequality, [10]:

$$0 \leq (\tfrac{1}{\sqrt{2}}Q^{\frac{1}{2}} \pm \sqrt{2}\Delta A^T PAQ^{-\frac{1}{2}})(\tfrac{1}{\sqrt{2}}Q^{\frac{1}{2}} \pm \sqrt{2}\Delta A^T PAQ^{-\frac{1}{2}})^T$$

$$= \tfrac{1}{2}Q \pm \Delta A^T PA \pm A^T P\Delta A + 2\Delta A^T PAQ^{-1}A^T P\Delta A. \tag{5}$$

From (5),

$$\tfrac{1}{2}Q - A^T P\Delta A - \Delta A^T PA \geq -2\Delta A^T PAQ^{-1}A^T P\Delta A, \tag{6}$$

hence

$$\tilde{Q} \geq \tfrac{1}{2}Q - 2\Delta A^T PAQ^{-1}A^T P\Delta A - \Delta A^T P\Delta A. \tag{7}$$

For the convergent property of $(A + \Delta A)$, it is sufficient that the right-hand side of (7) be positive definite. Q.E.D.

We may specialize this result to the case wherein A is in the phase-variable canonical form, to obtain the "largest" variation of the coefficients of a convergent polynomial. We first need, however, the following standard result from matrix theory.

THEOREM 3, [4]. Let M and K be n x n Hermitian matrices. If M is positive definite, then there exists an n x n matrix R for which $R^*MR = I$ and $R^*KR = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$. (Here * stands for transpose conjugate.) The number λ_j is a generalized eigenvalue satisfying $Kr^j = \lambda_j Mr^j$, $r^j \neq 0$, $j = 1,2,\ldots,n$. If K and M are real, then a real matrix R, with columns r^j, may be found such that $R^*MR = I$ and $R^*KR = \mathrm{diag}(\lambda_1,\ldots,\lambda_n)$, where λ_j's are the roots of $\det(\lambda M - K) = 0$, and I is an identity matrix.

THEOREM 4. Consider a monic-convergent polynomial $b(z) = \Sigma_{j=0}^n b_j z^{n-j}$. Let $b_j \to b_j + \Delta b_j$. Then there exists an upper bound β_j, where $\beta_j^2 = 1/2x_{nn}\lambda$, such that $b(z) = z^n + b_1 z^{n-1} + \ldots + (b_j + \Delta b_j)z^{n-j} + \ldots + b_n$, remains convergent if

$$(\Delta b_j)^2 \leq \beta_j^2. \tag{8}$$

Here x_{nn} is the (n,n)-entry of $(P + 2PAQ^{-1}A^T P)$, Q an arbitrary symmetric positive-definite matrix, and $P = P^T > 0$ is the unique solution of (1), for A in the phase-variable canonical form with $b(z) = \det(zI - A)$, and λ is the largest (or the unique nonzero) root of $\det(\lambda Q - K) = 0$, with $K = \mathrm{diag}(0,\ldots,0,\underset{\sim}{1}_j,0,\ldots,0)$. (Notice that by the asymptotic convergent property of $b(z)$, $x_{nn} \neq 0$.)

PROOF. Suppose $A \to A + \Delta A$, then, from Theorem 2, $A + \Delta A$ remains convergent matrix if $\Delta A^T(P + 2PAQ^{-1}A^T P)\Delta A \triangleq \Delta A^T X\Delta A < Q/2$. If $b_j \to b_j + \Delta b_j$, then $A \to A + \Delta A$, where ΔA has only one nonzero element $-\Delta b_j$ in its (n,j) entry. Substituting ΔA and X into (2), we get $(\Delta b_j)^2 x_{nn} K < Q/2$. Here $K = K^T = \mathrm{diag}(0,\ldots,0,1_j,0,\ldots,0) \geq 0$. By Theorem 3, Q and K can be simultaneously diagonalized, and therefore $(\Delta b_j)^2 x_{nn}\mathrm{diag}(\lambda_1,\ldots,\lambda_n) < I/2$. Q.E.D.

In many practical situations, given a characteristic polynomial b(z), the coefficients of this polynomial are $b_j = b_j(p)$, where p is a vector of system parameters. Therefore, whenever a parameter changes it affects more than one coefficient. The following theorem pertains to this case.

THEOREM 5. Consider the monic-convergent polynomial b(z) of Theorem 4. If a perturbation "Δp" causes all the coefficients of b(z) to change, then subject to the following upper-bound

$$\sum_{j=1}^{n} (\Delta b_j)^2 < \lambda_{min}(Q)/2x_{nn}, \tag{9}$$

the polynomial remains convergent. Here x_{nn} and Q are the same as those of Theorem 4, and $\lambda_{min}(Q)$ is the minimum eigenvalue of Q.

PROOF. Suppose $A \rightarrow A + \Delta A$ with ΔA has nonzero elements only on its last row. Then by Theorem 2, $(A + \Delta A)$ remains a convergent matrix if $\Delta A^T(P + 2PAQ^{-1}A^TP)\Delta A < Q/2$. As in Theorem 4, we assume $X \triangleq P + 2PAQ^{-1}A^TP$, then

$$\Delta A^T X \Delta A = x_{nn} \begin{bmatrix} (\Delta b_n)^2 & \Delta b_n \Delta b_{n-1}, \ldots, \Delta b_n \Delta b_1 \\ \Delta b_{n-1} \Delta b_n & (\Delta b_{n-1})^2, \ldots, \Delta b_{n-1} \Delta b_1 \\ \multicolumn{2}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ \Delta b_1 \Delta b_n & \Delta b_1 \Delta b_{n-1}, \ldots, (\Delta b_1)^2 \end{bmatrix} \triangleq x_{nn} \Pi < Q/2. \tag{10}$$

In this inequality every element is known a priori except the perturbed matrix Π, which in general is a nonlinear function of $(\Delta p)^2$--the quantity that is desired to be evaluated. In this case it is easy to see that the rank of Π is one, and therefore the maximum eigenvalue of Π is equal to its trace. If we let this maximum eigenvalue be less than or equal to the minimum eigenvalue of Q/2 (a conservative upper bound), then the inequality holds strongly. Thus we have $x_{nn}\mathrm{Tr}(\Pi) < \lambda_{min}(Q)/2$ or $x_{nn}\Sigma_{j=1}^{n}(\Delta b_j)^2 < \lambda_{min}(Q)/2$. Q.E.D.

3. COMPUTATIONAL CONSIDERATION: SINGLE PARAMETER VARIATION

The upper bounds developed in (2), (8) and (9) are functions of P and Q (which in turn P is a function of Q), and the following question arises naturally: for what "value" of $Q = Q^T > 0$, are these upper bounds the "largest" of their kind? Addressing this question we first elaborate on the word "largest." We must remember that in our analysis it is assumed that the eigenvalues of the system matrix A are sensitive to the parameter variations; otherwise there is no merit in further sensitivity analysis pertaining to system stability property. Considering this assumption in regard to the parameter sensitive, suppose ΔA has two nonzero entries $(\alpha\Delta p)_{kl}$ and $(\beta\Delta p)_{ij}$, where α and β are two scalars. Then by the "largest" upper bound on ΔA and correspondingly by the "largest" value of Δp it means that value for which some of the eigenvalues of the perturbed A, from the inside of the unit circle in the complex z-plane, are nearer to this circle than those of A or these eigenvalues are almost on the unit circle. Therefore any "additional" (in appropriate direction) change of p will result that the sensitive eigenvalue crosses the unit circle and causes oscillation, then unstability. In practical situation the unit circle is replaced by another circle of radius $|\delta| < 1$ in the z-plane. Larger δ results in more computations and perhaps greater ill-conditioning. Therefore the "largest" variation which preserves system stability is that value which brings some of the eigenvalues of the perturbed matrix not further than δ-circle in the z-plane.

The method of choosing the matrix Q so as to obtain this "largest" upper bound of ΔA on the first attempt is not known. The following theorem serves as a computational "tool" for finding the "largest" upper bound (of ΔA), for an arbitrary choice of matrix $Q = Q^T > 0$.

THEOREM 6. Let $Q = Q^T > 0$, A_0 be asymptotically convergent, and $P_0 = P_0^T > 0$ satisfies $A_0^T P_0 A_0 + Q = P_0$. Let ΔA be an arbitrary n x n matrix and rewrite the Lyapunov equation as

$$(A_0 + \varepsilon'\Delta A)^T P_0 (A_0 + \varepsilon'\Delta A) + Q$$

$$= P_0 + \varepsilon'(\Delta A^T P_0 A_0 + A_0^T P_0 \Delta A) + \varepsilon'^2 \Delta A^T P_0 \Delta A. \tag{11}$$

Then the following can be shown.

(i) There is a largest $|\varepsilon_0'| < 1$ such that

$$\Pi_0 \triangleq Q - \varepsilon_0(\Delta A^T P_0 A_0 + A_0^T P_0 \Delta A) - \varepsilon_0^2 \Delta A^T P_0 \Delta A > 0, \text{ if } |\varepsilon_0| < |\varepsilon_0'| < 1.$$

(ii) Let $A_1 = A_0 + \varepsilon_0 \Delta A$. If A_1 remains asymptotically convergent, then $A_1^T P_1 A_1 + Q = P_1$; and like (i), an $|\varepsilon_1'| < 1$ can be found such that

$$\Pi_1 \triangleq Q - \varepsilon_1(\Delta A^T P_1 A_1 + A_1^T P_1 \Delta A) - \varepsilon_1^2 \Delta A^T P_1 \Delta A > 0, \text{ if } |\varepsilon_1| < |\varepsilon_1'| < 1.$$

(iii) Similarly ε_0, ε_1, $\varepsilon_2, \ldots$, can be found. It can be shown that if $\theta = \Sigma_{i=0}^{\infty} \varepsilon_i$, then $(A_0 + \theta\Delta A)$ is not asymptotically convergent, but $A_0 + \varepsilon\Delta A$ is asymptotically convergent for $|\varepsilon| < |\theta|$.

PROOF.

(i) From $|\varepsilon_0'| < 1$ and Theorem 2, we have

$$\varepsilon_0^2 \Delta A^T (P + 2PAQ^{-1}A^T P)\Delta A < Q/2. \tag{12}$$

From (12), we have $-\varepsilon_0^2 \Delta A^T P \Delta A > -\frac{Q}{2} + \gamma Q$, where $\gamma = (2\varepsilon_0^2 \Delta A^T PAQ^{-1}A^T P \Delta A)Q^{-1}$; we let $\varepsilon_0 = 1$ in order to compute γ. This, of course, results in a conservative upper bound of ε_0 to be evaluated next. Thus:

$$\Pi_0 > \Pi_0' \triangleq (\frac{I}{2} + \gamma)Q - \varepsilon_0(A_0^T P_0 \Delta A + \Delta A^T P_0 A_0). \tag{13}$$

From (13), and Theorem 3, we can see that by simultaneous diagonalization of $(\frac{I}{2} + \gamma)Q > 0$ and the symmetric matrix $(A_0^T P_0 \Delta A + \Delta A^T P_0 A_0)$, we can find an ε_0 such that Π_0' and therefore Π_0 becomes a positive-definite matrix. We must realize that the possibility of having both $\pm\varepsilon_0$ in certain problems exists. That means perturbation to A_0 in both directions can be applied. These special cases are to be considered carefully in order to develop a corresponding computer program.

(ii) Straightforward proceedings of the result of step (i) will prove this part.

(iii) Continuing similarly as in step (i) above, we can show that by the appropriate choice of ε_k a sequence of matrices can be established such that at each stage $A_{k+1} = A_k + \varepsilon_k \Delta A$ remains asymptotically convergent; i.e., $\Pi_k = Q - \varepsilon_k(\Delta A^T P_k A_k + A_k^T P_k \Delta A) - \varepsilon_k^2 \Delta A^T P_k \Delta A > 0$. Each A_{k+1} has some eigenvalues distributed from the inside of the unit circle closer to this circle than those of A_k. As k increases, say k = m, and as we generate Π_m and A_{m+1}, we reach to a limiting case wherein A_{m+1} is no longer convergent or Π_m is no longer positive definite. In fact, because of numerical problems this limiting case occurs earlier than the true mathematical response. In practical situations as stated before the unit circle is replaced by a circle of radius less than one; we called this circle δ-circle. Because the initial matrix A_0 is assumed parameter sensitive, therefore, after a certain number of iterations the eigenvalues of perturbed A_0 cross this δ-circle and the computation stops.

The key point of the present theorem, however, is that A_{k+1} can be written as $A_{k+1} = A_{k-1} + \varepsilon_{k-1}\Delta A + \varepsilon_k \Delta A$ or simply, k times repetition of the latter yields $A_{k+1} = A_0 + \Sigma_{i=0}^{k} \varepsilon_i \Delta A \triangleq A_0 + \theta_k \Delta A$. This completes the proof of this theorem and the requirement for the convergence of the algorithm presented later.

From the above theorem we conclude that by choosing any $Q = Q^T > 0$ as a starting point, we may develop an iterative scheme in order to determine the upper bound or the "largest" perturbation of parameters of a sensitive convergent matrix. For simplifying of our analysis we choose Q = I.

EXAMPLE 1. Suppose

$$\Delta A = \begin{pmatrix} & 0 & \\ 0 & (\Delta p)_{ij} & 0 \\ & 0 & \end{pmatrix} . \tag{14}$$

Then, from (2), we have

$$(\Delta p)^2 \begin{pmatrix} & 0 & \\ 0 & 1_{ji} & 0 \\ & 0 & \end{pmatrix} (P + 2PAQ^{-1}A^T P) \begin{pmatrix} & 0 & \\ 0 & 1_{ij} & 0 \\ & 0 & \end{pmatrix} < Q/2,$$

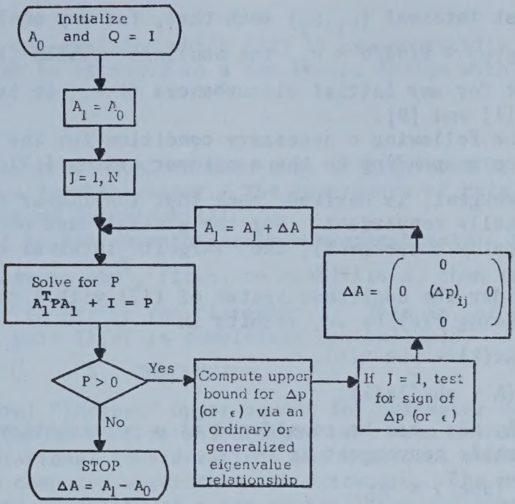
$$(\Delta p)^2 \begin{pmatrix} & 0 & \\ 0 & (x_{jj})_{ii} & 0 \\ & 0 & \end{pmatrix} \triangleq (\Delta p)^2 K < Q/2 \tag{15}$$

where x_{jj} is the (j,j)-entry of $P + 2PAQ^{-1}A^T P$. From (15) and using Theorem 3, we have $(\Delta p)^2 < 1/2\lambda$, where λ is the largest (or the unique nonzero) root of $\det(\lambda Q - K) = 0$. If we let Q = I, then this upper bound for parameter variations which does not change system stability becomes a simple ordinary eigenvalue of K which is x_{jj}.

Regardless of the choices of the ordinary or any generalized eigenvalues, the upper bounds developed in this paper are local, i.e., they depend on the choice of Q matrix; following algorithm pertains to this case.

ALGORITHM 1. DETERMINATION OF THE LARGEST PERTURBATION

Following flow-chart illustrates the basic idea behind the computation of the largest (Δp), for a perturbed matrix ΔA that is a linear function of Δp, for which ($A + \Delta A$) remains convergent. Here we replace the unit-circle by a δ-circle.



Initialize A_0 and $Q = I$
$A_1 = A_0$
$J = 1, N$
$A_1 = A_1 + \Delta A$
Solve for P $A_1^T P A_1 + I = P$
$\Delta A = \begin{pmatrix} & 0 & \\ 0 & (\Delta p)_{ij} & 0 \\ & 0 & \end{pmatrix}$
$P > 0$
Yes
No
Compute upper bound for Δp (or ϵ) via an ordinary or generalized eigenvalue relationship
If $J = 1$, test for sign of Δp (or ϵ)
STOP $\Delta A = A_1 - A_0$

Figure 1. Schematic Diagram of Flow-Chart for Algorithm 1.

All the preceding derivations were centered around ΔA as a linear function of Δp. Suppose ΔA is not so; i.e., let as an example

$$A = \begin{pmatrix} p^2 & 1 \\ p & 0 \end{pmatrix}, \text{ then } \Delta A = \begin{pmatrix} 2p\Delta p + \Delta p^2 & 0 \\ \Delta p & 0 \end{pmatrix}. \tag{16}$$

In this case, though the sufficient condition for $A + \Delta A$ remains asymptotically convergent is (2), but computation of Δp is more complicated than the previous case. However, for certain special cases, in which the left-hand side of (2) has some additional algebraic properties, though not always easily observed, (2) may lead to some computational simplicity as has been the case in Theorem 5. Again, this derivation must be updated by applying an Algorithm as that of Algorithm 1. The study of the case in which ΔA is a function of several parameters, and these parameters change simultaneously (though again this ΔA satisfies (2) as a sufficient condition), is more complicated than the present analysis and that study is the subject of a separate report.

4. CERTAIN EXTENSIONS TO THE NONLINEAR SYSTEMS

In closing the contents of this paper, the following necessary condition for absolute stability of a set of nonlinear difference equations is derived. For the importance and the historical background of this problem one may refer to [1], [5], [8] and [9].

Our problem is as follows.

Consider a set of nonlinear difference equations described by

$$x(k+1) = Ax(k) + bf(\sigma),$$
$$\sigma(k) = c^T x(k); \tag{17}$$

where $x(k)$ is a vector; A is an n x n constant matrix; $f(\sigma)$ is a scalar function of σ; and b and c are vectors. Replacing $f(\sigma)$ by a linear function $\mu\sigma$(μ is a scalar), results in a system which is asymptotically convergent for all k in an interval $[\mu_1, \mu_2]$. Then the problem is to find the largest interval $[\mu_1,\mu_2]$ such that, for any nonlinear function $f(\sigma)$ satisfying $\mu_1 < f(\sigma)/\sigma < \mu_2$, the nonlinear system (1) is asymptotically convergent for any initial disturbances (i.e., it is convergent in the "large") [1] and [9].

In the following a necessary condition for the largest interval $[\mu_1,\mu_2]$, corresponding to the nonlinear system (17), with A being asymptotically convergent, is derived, such that the linear truncated system remains asymptotically convergent. For the general case of A (not necessarily asymptotically convergent), the "largest" interval can be similarly derived.

Consider the nonlinear system of (17) with A convergent and real. Then replacing $f(\sigma)$ by $\mu\sigma$, results in

$$\begin{aligned} x(k+1) &= Ax(k) + \mu bc^T x(k), \\ &= (A + \mu bc^T)x(k). \end{aligned} \tag{18}$$

For this A, μbc^T can be considered as a perturbation, then $(A + \Delta A)$ remains asymptotically convergent if

$$(\mu bc^T)^T\ X\ (\mu bc^T) < \tfrac{1}{2}Q, \tag{19}$$

where $X \triangleq P + 2PAQ^{-1}A^TP$. From (19) and the fact that the rank of $[(bc^T)^T X(bc^T)]$ is one, we see that (19) requires that either

$$\mu^2 < \frac{\lambda_{\min}(Q)}{2\mathrm{Tr}[(bc^T)^T X(bc^T)]}\quad , \tag{20}$$

which is a conservative upper bound, or

$$\mu^2 < 1/2\lambda \quad , \tag{21}$$

which λ is the unique nonzero generalized eigenvalue of $(bc^T)^T X(bc^T)$ with respect to Q. If we let $Q = I$, this λ becomes a simple ordinary eigenvalue of $[(bc^T)^T X(bc^T)]$. The two upper bounds (20) and (21) are both local, since these depend on the initial choice of Q. A convergent iterative scheme, supported by Theorem 6, as that of Algorithm 1, or by simultaneous diagonalization, using (19), can be developed in order to derive the "largest" μ, such that $(A + \mu bc^T)$ remains asymptotically convergent. This interval which is found iteratively by an Algorithm similar to Algorithm 1, is the necessary condition on μ for $(A + \mu bc^T)$ to be convergent.

The problem is now manageable as follows.

ALGORITHM 2. Consider the set of nonlinear difference equations

$$\begin{aligned} x(k+1) &= Ax(k) + bf(\sigma), \\ \sigma(k) &= c^T x(k) \end{aligned} \tag{22}$$

with A real and convergent. Then replacing $f(\sigma)$ by $\mu\sigma$ results in $x(k+1) = (A + \mu bc^T)x(k)$. For the largest $\mu \varepsilon [\mu_1,\mu_2]$ that iteratively computed as described earlier in this section, $(A + \mu bc^T)$ remains asymptotically convergent. Then it is to be tested for what $[\underline{\mu}_1,\bar{\mu}_2]$ following is true

$$\underline{\mu}_1 < f(\sigma)/\sigma < \bar{\mu}_2, \quad \text{for } x\ \varepsilon\ \chi. \tag{23}$$

where χ is either the whole space or it is a locally closed subspace around the operating point. Then

$$\mu\varepsilon[\mu_1,\mu_2]\bigcap[\underline{\mu}_1,\bar{\mu}_2], \tag{24}$$

is the wanted "largest" interval for which (22) is asymptotically convergent. Algorithm 2 can be extended to a non-linear system with many nonlinearities such as

$$x(k+1) = Ax(k) + BF(\sigma). \tag{25}$$

However, this analysis is more involved than that of Algorithm 2. This extension is subject of a further report. The importance of this problem has been recently emphasized by Gaiduk [5].

For the case that A is not asymptotically convergent matrix then it is not difficult to choose an μbc^T, first, to stabilize A, then to perturb this $(A + \mu bc^T)$ in order to derive the "largest" μ. This is, of course, true if and only if the pair (A,b) is completely controllable.

5. CONCLUSION

In this paper several "largest" upper bounds for parameter variations of the convergent matrices as well as polynomials have been developed. The corresponding computational considerations for numerical evaluation of these upper bounds with computer algorithms are discussed. The necessary condition of the absolute stability of a set of nonlinear difference equations has been also advanced. These results have important applications in digital system design.

Extensions of these results to systems with several parameters variations and to the nonlinear systems with several nonlinearities are subject of a further study.

REFERENCES

[1] M. A. Aizerman, "Some Unsolved Problems in the Theory of Automatic Control and Fuzzy Proofs," (Tech. Notes and Corresp.) IEEE-Trans. on Auto. Control, Vol. AC-22, No. 1, pp. 116-118, February 1977.

[2] Yu. N. Andreev, "Algebraic Methods of State Space and Linear Object Control Theory (Survey of Foreign Literature)," Automation and Remote Control, Vol. 38, No. 3, pp. 305-342, March 1977.

[3] M. Eslami, "Sensitivity Analysis and Synthesis in Automatic Control Systems," Ph.D. Dissertation, The University of Wisconsin-Madison, Madison, Wisconsin, May 1978.

[4] J. N. Franklin, Matrix Theory. Englewood Cliffs, New Jersey: Prentice-Hall, 1968.

[5] A. I. Gaiduk, "Absolute Stability of Control Systems With Several Nonlinearities," Automation and Remote Control, Vol. 37, No. 6, pp. 815-821, November 1976.

[6] R. E. Kalman and J. E. Bertram, "Control System Analysis and Design Via the 'Second Method' Lyapunov, I. Continuous-Time Systems," Trans. of the ASME-Journal of Basic Engineering, Vol. 82, Series D, No. 2, pp. 371-393, 1960.

[7] B. C. Kuo, Digital Control Systems. Champaign, Illinois: SRL Publishing Co., 1977.

[8] K. S. Narendra and J. H. Taylor, Frequency Domain Criteria for Absolute Stability. New York: Academic Press, 1973.

[9] E. S. Pyatnytsky, "New Research on Absolute Stability of Automatic Control Systems," (Review) Automation and Remote Control, Vol. 29, No. 6, pp. 855-881, June 1968.

[10] D. L. Russell, Mathematics of Finite-Dimensional Control Systems, theory and design. New York: Dekker, 1979.

[11] B. A. Shane and S. Barnett, "Sensitivity of Stable Linear Systems," (Tech. Notes and Corresp.) IEEE-Trans. on Auto Control, Vol. AC-17, No. 1, pp. 148-150, February 1972.

[12] ________, "Sensitivity of Convergent Matrices and Polynomials," Journal of Math. Analysis and Application, Vol. 43, pp. 114-122, 1973.

[13] ________, "Insensitivity of Constant Linear Systems to Finite Variations in Parameters," Recent Mathematical Developments in Control, Ed. D. J. Bell, Academic Press, pp. 393-404, 1973.

[14] ________, "Insensitivity of Asymptotically Stable Matrices to Variations in Submatrices," Int. Journal of Systems Science, Vol. 6, No. 7, pp. 621-632, 1975.

QUANTIZATION-PERMUTATION CODES AND BUFFER-ADAPTED HUFFMAN CODES†

TOBY BERGER, MING UN CHANG and SUI-YIN TUNG-KLEINBERG
T. Berger, School of Electrical Engineering, Cornell University, Ithaca, NY 14853. M. Chang and S.-Y. Tung-Kleinberg, formerly with the School of Electrical Engineering, are with Hughes Aircraft, Space and Communications Group, Los Angeles, CA, 90009.

Two source digitization techniques are compared--(1) quantization-permutation (QP) codes and (2) buffer-adapted Huffman (BAH) codes. The QP technique is a block coding scheme, whereas the BAH technique is variable-length.

Before discussing QP encoding we briefly describe permutation source codes. A blocklength n permutation code consists of all $n!/\prod_{k=1}^{m} n_k!$ distinct permutations of a basic word of the form

$$(\underbrace{\mu_1,\mu_1,\ldots,\mu_1}_{n_1},\underbrace{\mu_2,\mu_2,\ldots,\mu_2}_{n_2}\cdots\underbrace{\mu_m,\mu_m,\ldots\mu_m}_{n_m})$$

The integers n_k, where $n_1 + n_2 + \cdots + n_m = n$, are called the group sizes, and the numbers μ_k , $\mu_1 < \mu_2 < \cdots < \mu_m$, are called the reconstruction levels. It is easy to show that, for a broad class of convex distortion measures including the squared error criterion, optimum encoding of a source block $\underline{x} = (x_1,\ldots,x_n)$ by means of a permutation code is accomplished by replacing the n_1 smallest x_i's by μ_1, the n_2 next smallest by $\mu_2,\ldots$, and the n_m largest x_i's by μ_m.

The first step of QP encoding is to quantize finely each letter x_i of the source block $\underline{x} = (x_1,\ldots,x_n)$. The quantization thresholds are bunched in triplets near the midpoints between successive reconstruction levels of the underlying permutation code, thereby forming upper and lower guard bands. In the second step the letters x_i that fell in the guard bands are reassigned to the neighboring main bands such that the final number in the kth main band is exactly the size n_k of the kth group of the permutation code.

For large n the number of letters that fail to get encoded by QP in precisely the same way they would were the permutation code used without quantization is $O(\sqrt{n})$ with high probability. Moreover, most of these get moved to an immediately neighboring group. Hence, the quantization step introduces only small additional encoding distortion yet completely eliminates the need to order the data samples. (Total ordering would require $O(n \log n)$ comparisons, although a partial ordering would suffice.)

---

† This work was supported in part by NSF Grants GK-32240, GK-41200, ENG 76 19799 and ECS-7921976.

With the aid of the analytical formula for the distortion of QP codes we now present, design tradeoffs can be conducted concerning the proper relationship between $\{n_k\}$, $\{\mu_k\}$ and the widths of the guard bands. The expected distortion of a QP code is $E(n^{-1}\sum_{ij}\sum N_{ij}d_{ij})$, where N_{ij} is the random number of source samples that fall in the ith bin$(a_i, a_{i+1}]$ but get transferred to the j^{th} bin by the QP procedure, and d_{ij} is the average distortion that results each time such a transfer occurs. The index i ranges over both the guard bands and the main bands while j ranges only over the main bands. It is easy to see that

$$d_{ij} = \int_{a_i}^{a_{i+1}} (x-\mu_j)dF(x)/\,p_i$$

where F(x) is the source distribution, $p_i = \int_{a_i}^{a_{i+1}} dF(x)$ and μ_j is the mean of the j^{th} main band with its upper and lower guard bands included. To complete the evaluation of the MSE, we need a formula for

$E\,N_{ij} = \sum_{\ell=1}^{n_j} \ell P(N_{ij} = \ell)$, so we must evaluate $P(N_{ij} = \ell)$. Toward this end we

define $s_j = \sum_{i \leq j} n_i$, $\pi_k = \sum_{i \leq k} p_i$ and $\lambda_i = p_i/\pi_i$

and let $b(n,m,p) = \binom{n}{m}p^m(1-p)^{n-m}$. Analysis then reveals that for $1 \leq \ell < n_j$,

$$\begin{aligned}P(N_{ij} = \ell) = {} & b(n,s_{j-1}+\ell,\pi_i) \cdot \sum_{h \geq \ell} b(s_{j-1}+\ell,h,\lambda_i) \\ & + \sum_{h=\ell+1}^{n_{j-1}} b(n,s_{j-1}+h,\pi_i) \cdot b(s_{j-1}+h,\ell,\lambda_i) \\ & + b(n,s_j-\ell,\pi_{i-1}) \cdot \sum_{h \geq \ell} b(n-s_j+\ell,h,p_i/1-\pi_i)\end{aligned}$$

and

$$P(N_{ij} = n_j) = \sum_{h=0}^{s_{j-1}} (b(n,h,\pi_{i-1}) \cdot \sum_{k=s_j-h}^{n-h} b(n-h,k,p_i/1-\pi_i)).$$

Despite the apparent clumsiness of these formulas, we had no difficulty evaluating them for $n = 1000$, 10 main bins and 20 guard bins; Stirling approximations to binomial terms were used where appropriate.

Quantization followed by variable-length encoding of the quantizer outputs is another popular method of source digitization. A Huffman variable-length code usually is employed. Because the resulting code words are of different lengths, a buffer must be used. This leads to buffer overflow and buffer underflow problems. The overflows result in the loss of certain data samples with a concomitant increase in average distortion. The underflows result in an effective increase in the encoding rate by virtue of one's having to stuff zeros into the buffer.

The buffer overflow problem can be eliminated altogether at the expense of a modest increment in distortion. This is accomplished by using a short code word that does not cause an overflow whenever the source produces an unlikely letter whose long code word would have resulted in an overflow. In particular, in such a circumstance one uses that Huffman code word from those which do fit in the buffer whose reconstruction level is closest to the source output. We refer to this technique as buffer-adapted Huffman (BAH) encoding.

The performance of BAH codes has been analyzed by means of a finite-state Markov chain model for the excursions of the buffer pointer. Equilibrium analysis of this model yields expressions for the average distortion and the encoding rate. Both overflow prevention and bit stuffing are taken into account.

Computations performed for the Gaussian-MSE case indicate that the BAH technique is simpler to implement than the QP technique for the same rate-distortion performance. In Figure 1 the performances achieved by the QP and BAH techniques are compared with exact permutation coding, with entropy-coded uniform quantization and with Shannon's rate-distortion function. At each coding rate we show QP with no bin splitting, with bin bisection, with bin trisection (the guard band technique described herein), and with bin quadrusection. Finer sectioning brings the performance more closely into accord with exact permutation coding. Although the BAH technique (called B-VL in the figure) outperforms QP with quadrusection, it has a difficulty that can be highly troublesome in practice. Because of its asynchronous nature, it is important to adjust the quantization thresholds such that the average word length $\bar{m}$ is slightly less than the average number m of symbols extracted from the buffer per cycle. This yields good performance in that, although underflow occurs fairly frequently and requires bit stuffing with some concomitant rate loss, overflow almost never occurs. If, however, the source distribution is not known well, then the actual letter probabilities may differ from their design values in such a way as to make $\bar{m} > m$. In that event the buffer pointer will spend most of the time near the overflow end, resulting in relatively frequent overflows and therefore in significantly increased average distortion. Accordingly, it is necessary in practice to design with a margin of safety in the direction $\bar{m} < m$.

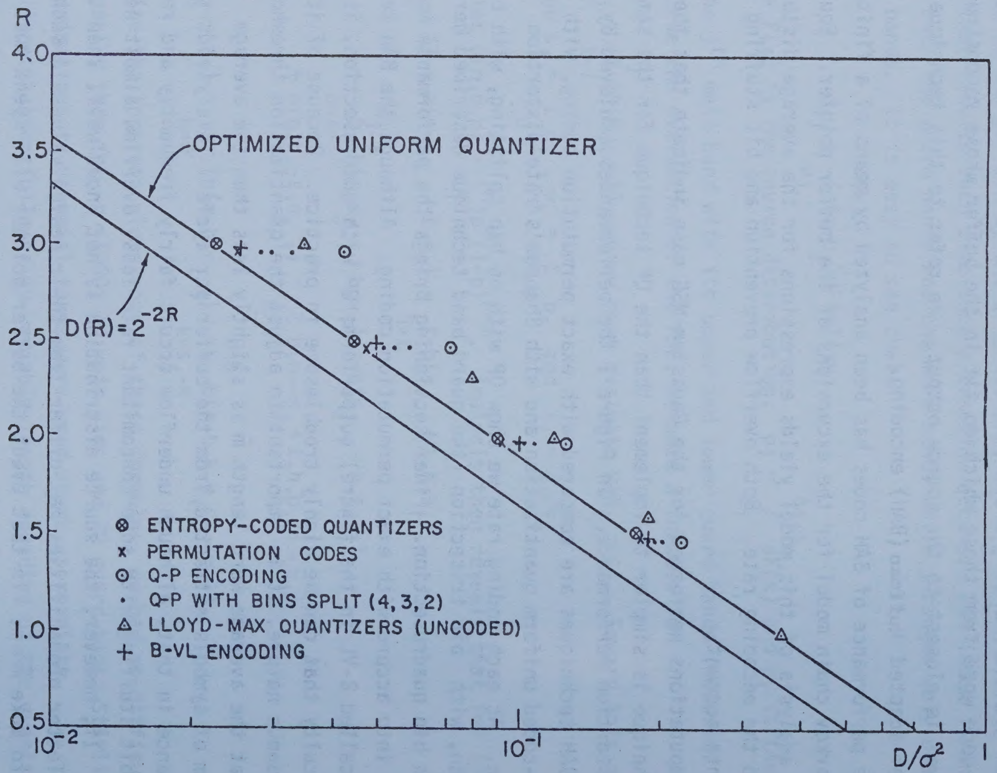

R
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
OPTIMIZED UNIFORM QUANTIZER
D(R)=2^{-2R}
⊗ ENTROPY-CODED QUANTIZERS
× PERMUTATION CODES
⊙ Q-P ENCODING
· Q-P WITH BINS SPLIT (4, 3, 2)
△ LLOYD-MAX QUANTIZERS (UNCODED)
+ B-VL ENCODING
10^{-2}
10^{-1}
1
D/σ²

Figure 1

TREE-SEARCHED BLOCK SOURCE CODES*

ROBERT M. GRAY
Stanford University
Stanford, California

SUMMARY

Tree-searched vector quantizers or block source codes subject to a fidelity criterion were first introduced in Buzo, et al [1]. These codes permit a compromise between performance and implementation complexity by reducing the number of searches and distortion computations at the expense of additional distortion. A tree-searched block code views a source vector and determines an output vector in a codebook of size 2**R by performing a sequence nearest neighbor decisions from small codebooks of key words. For example, a binary tree-searched code would make R binary comparisons to obtain a final codeword as opposed to the 2**R comparisons required for a full searched code. Unlike traditional tree encoding systems, however, the entire source vector is used for each comparison and only one path through the tree is followed.

In this talk several preliminary results will be described concerning the optimality properties, tradeoffs between performance and implementation complexity, and computer-aided design alternatives for these codes. Except for some very special cases, there are still no Shannon-style coding theorems for this code structure, yet such codes have been found to perform quite well in experiments on highly correlated data such as human speech.

[1]. A. Buzo, A.H. Gray, Jr., R.M. Gray, and J.D. Markel, "Speech coding based on vector quantization," IEEE Trans. ASSP, to appear.

*Research supported by the Air Force Office of Scientific Research under contract #F49620-79-C-0058.

VARIABLE-LENGTH TO VARIABLE-LENGTH ENCODERS ARE ASYMPTOTICALLY MEAN STATIONARY

JOHN C. KIEFFER
Dept. of Math., Univ. of MO
Rolla, MO 65401

JAMES G. DUNHAM
Dept. of Elec. Engr., Washington Univ.
St. Louis, MO 63130

Abstract. A wide class of source encoders is defined which includes the variable-length to variable-length encoders, finite-state encoders, and many others of practical interest. A simple proof is given that any encoder in the class encodes asymptotically mean stationary (a.m.s.) sources into a.m.s. sources.

Let A^∞, B^∞ be the one-sided sequence spaces on the finite alphabets A, B. Consider a quintuple $(S, \{L_s : s \in S\}, f, g, s_0)$, where S is a finite set, each L_s is a bounded measurable map from $A^\infty \to \{1,2,\ldots\}$, f is a measurable map from $S \times A^\infty \to S$, g is a measurable map from $S \times A^\infty$ onto some finite set of variable-length sequences from B, and s_0 is a fixed element of S. Given such a quintuple, construct a code $\phi: A^\infty \to B^\infty$ in the following way. Fix an arbitrary $x^{(0)} \in A^\infty$ and define $x^{(1)}, x^{(2)}, \ldots \in A^\infty$ and $s_1, s_2, \ldots \in S$ so that

$$x^{(i)} = T^{L_{s_{i-1}}(x^{(i-1)})}(x^{(i-1)}), \quad s_i = f(s_{i-1}, x^{(i-1)}), \quad i \geq 1,$$

where T is the shift on A^∞. Then $\phi(x^{(0)}) \in B^\infty$ is the concatenation of $g(s_0, x^{(0)})$, $g(s_1, x^{(1)}), \ldots$. Let C be the class of all codes ϕ constructed in this way. One could argue that C contains any implementable source encoder. For example, if S is a singleton, ϕ is a variable-length to variable-length encoder. If all the L_s are identically one and g maps into B, one obtains the finite-state encoders.

A source encoder (i.e., a measurable map from $A^\infty \to B^\infty$) is a.m.s. if it carries (one-sided) a.m.s. sources on the alphabet A into a.m.s. sources on B. (See [1] for a treatment of a.m.s. sources and processes.)

Our main result is the following

Theorem. Every encoder in C is a.m.s.

As a special case, we get the title of this paper.

Definition. We say a process $\{Z_i\}_{i=1}^\infty$ is a sampling of a process $\{X_i\}_{i=1}^\infty$ if there is an event E such that

$$\limsup_{n\to\infty} n^{-1} \sum_{i=1}^{n} I_{\{X_i \in E\}} > 0 \quad \text{a.s.}$$

and $Z_i = X_{T_i}$ $(i \geq 1)$, where $T_1, T_2, \ldots$ are the times such that $X_{T_i} \in E$.

We omit the easy proof of the following

Sampling Lemma. Any sampling of an a.m.s. process is also a.m.s.

Proof of Theorem. Fix $\phi \in C$ and the quintuple $(S,\{L_s\}, f, g, s_0)$ which gives rise to ϕ as described earlier. Let $\hat{B}$ be the finite set of variable-length sequences from B which is the range of g. Let τ be an element not in $\hat{B}$. Consider the encoder from $A^\infty \to (\hat{B} \cup \{\tau\})^\infty$ which carries $x^{(0)} \in A^\infty$ into

$$(g(s_0,x^{(0)}), \tau,\ldots,\tau, g(s_1,x^{(1)}), \tau,\ldots,\tau, g(s_2,x^{(2)}),\ldots),$$

where each entry $g(s_i,x^{(i)})$ is followed by $L_{s_i}(x^{(i)}) - 1$ entries equal to τ. It is easy to see that this encoder is finite-state. Finite-state encoders are a.m.s. [2]. Therefore, by the sampling lemma, the encoder $A^\infty \to \hat{B}^\infty$ which carries $x^{(0)}$ into

$$(g(s_0,x^{(0)}), g(s_1,x^{(1)}),\ldots),$$

is a.m.s. The encoder from $\hat{B}^\infty \to B^\infty$ obtained by concatenating is a.m.s. [1]. Therefore ϕ is a.m.s.

[1] R. M. Gray and J. C. Kieffer (1980), "Asymptotically Mean Stationary Measures," to appear.

[2] J. C. Kieffer and M. Rahe, (1978), "Markov Channels are Asymptotically Mean Stationary," to appear.

THE PRINCIPLE OF CONSERVATION OF ENTROPY*

JAMES G. DUNHAM
Department of Electrical Engineering
Washington University
St. Louis, Missouri 63130

ABSTRACT

The Principle of Conservation of Entropy states that the entropy rate of the output process of a noiseless source code is related to the entropy rate of the source input process by the average length of the code. For the class of variable-length to variable-length codes, a proof is given for a class of stationary and ergodic sources which includes Markov sources.

1. INTRODUCTION

Let X be a stationary and ergodic source and denote by $\bar{H}(X)$ the mean entropy rate of X. Connecting X to a noiseless code C yields the encoded message Y. The Principle of Conservation of Entropy states that

$$\bar{H}(Y) = \frac{\bar{H}(X)}{\bar{L}} \tag{1}$$

where $\bar{L}$ is average code length. Intuitively, any uniquely decipherable coding should yield an encoded message which satisfies (1). A converse to the noiseless source coding theorem is easily derived from (1).

A code is called a fixed-length to variable-length (FL-VL) code if a fixed-number of source symbols are encoded into a possibly variable number of code symbols. The Huffman code is an example of FL-VL code. A code is called a variable-length to variable-length (VL-VL) code if a possibly varying number of source symbols are encoded into a possibly varying number of code symbols. Both run-length and Huffman codes are examples of a VL-VL code. A finite-state code is a code representable by a finite-state machine. A code is called monotone code if the codes are monotonic under the natural partial ordering on finite strings.

The first rigorous proof of the Principle of Conservation of Entropy was given by Katona and Tusnady [1] for FL-VL codes and stationary and ergodic sources. Jacobsen [2] later gave a simpler proof. Sidel'nikov [3]

---

*This work was supported by NSF Grant ENG-7910001.

gave a proof for finite-state codes and Markov processes. Csiszar, Katona and Tusnady [4] gave a proof for monotone codes and sources where the average code length converges in probability to $\bar{L}$. We note that FL-VL codes are monotone codes and that there are finite-state codes which are not monotone codes.

A VL-FL code is not necessarily a monotone code but it is a finite-state code. We establish a Principle of Conservation of Entropy for VL-VL codes and a class of stationary and ergodic sources which includes Markov sources.

2. VL-VL CODES

A full binary tree is a binary tree in which all paths through the tree terminate at a terminal node. The number of terminal nodes is called the size of the tree. If all paths through a full binary tree are of equal length from the root node, then it is called a fixed-length (FL) tree, else it is called a variable-length (VL) tree. A variable-length to variable-length (VL-VL) code consists of a pair (T_s, T_c) of back-to-back full binary trees of equal size. T_s is called the source coder or parsing tree and T_c is called the channel coder or parsing tree. If $T_s(T_c)$ is a FL tree, then the code is a FL-VL (VL-FL) code. We note that FL-VL codes correspond to the usual Huffman codes. Jelinek and Schneider [5] discuss VL-FL codes and Cohn and Mulder [6] discuss VL-VL codes.

Let X be a stationary and ergodic source with probability measure μ and let (T_s, T_c) be an invertible VL-VL code. Placing X into T_s yields an intermediate process Z with a probability measure defined by

$$\nu(G) = \mu(T_s^{-1}G).$$

Placing Z into T_c yields the encoded message process Y with a probability measure defined by

$$\sigma(G) = \nu(T_c^{-1}G).$$

The lemma below shows that if X is Markov then Z is also Markov. In particular, note that Z will have a finite number of irreducible classes and hence a finite number of ergodic submodes.

LEMMA: Let X be a Markov source and let T be a VL-FL code. Then Z is a Markov source.

3. MAIN RESULT

The main result is the following theorem. The first corollary follows from the lemma. The second corollary is the converse to the noiseless source coding theorem.

THEOREM: Let X be a stationary and ergodic source and let (T_s, T_c) be an invertible VL-VL code. If the intermediate process Z has a finite number of ergodic submodes θ_i, i = 1,..., n, then the Principle of Conservation of Entropy holds with

$$\bar{H}(Y) = \left[\sum_{i=1}^{n} \frac{L_s(\theta_i)}{L_c(\theta_i)} p(\theta_i) \right] \bar{H}(X)$$

where $L_s(\theta_i)$ $(L_c(\theta_i))$ are the average code length of placing Z through the FL-VL coder $T_s^{-1}(T_c)$.

COROLLARY 1: If X is Markov, then the Principle of Conservation of Entropy holds with

$$\bar{H}(Y) = \bar{L}\bar{H}(X).$$

COROLLARY 2: For any invertible (VL-VL) code and Markov source we have

$$\bar{L} \geq \frac{\bar{H}(X)}{\log ||A||}$$

where $||A||$ denotes the size of the code alphabet.

PROOF OF THEOREM: Let $\bar{H}(\eta)$ denote the mean entropy rate of the probability measure η. It follows from Kieffer and Dunham [7] that ν will be an asymptotically mean stationary (a.m.s.) process with stationary mean $\bar{\nu}$. Since $\bar{\nu}$ has a finite number of ergodic submodes, the ergodic decomposition of $\bar{\nu}$ is

$$\bar{\nu}(G) = \sum_{i=1}^{n} \bar{\nu}_{\theta_i}(G) p(\theta_i).$$

Gray and Kieffer [8] show that

$$\bar{H}(\nu) = \sum_{i=1}^{n} \bar{H}(\bar{\nu}_{\theta_i})\nu(\theta_i)$$

$$= \sum_{i=1}^{n} \bar{H}(\bar{\nu}_{\theta_i})p(\theta_i) = \bar{H}(\bar{\nu}).$$

Defining

$$\nu_\theta(G) = \frac{\nu(G \cap \theta)}{\nu(\theta)}$$

it is easy to show that ν_θ is an ergodic a.m.s. process.

Consider placing ν_θ through T_s^{-1}, the inverse to T_s. This induces a probability measure

$$\eta_\theta(G) = \nu_\theta((T_s^{-1})^{-1}G) = \mu_{T_s^{-1}\theta}(G)$$

where

$$\eta(G) = \sum_{i=1}^{n} \eta_{\theta_i}(G)p(\theta_i)$$

$$= \sum_{i=1}^{n} \mu_{T_s^{-1}\theta_i}(G)\mu(T_s^{-1}\theta_i)$$

$$= \mu(G),$$

that is, we recover the original probability measure μ since $T_s^{-1}T_s$ is the identity mapping.

Katona and Tusnady [1] show for the FL-VL code T_s^{-1} that

$$\bar{H}(\eta_\theta) = \frac{\bar{H}(\nu_\theta)}{\bar{L}_s(\theta)} \tag{2}$$

where $\bar{L}_s(\theta)$ is the average code length through T_s^{-1}. Gray and Kieffer [8] show that η_θ is an a.m.s. source. Hence

$$\bar{\eta}(G) = \lim_{k\to\infty} k^{-1} \sum_{j=0}^{k-1} \left[\sum_{i=1}^{n} \mu_{T_s^{-1}\theta_i}(T^jG)p(\theta_i) \right]$$

$$= \sum_{i=1}^{n} p(\theta_i) \left[\lim_{k\to\infty} k^{-1} \sum_{j=0}^{k-1} \mu_{T_s^{-1}\theta_i}(T^jG) \right]$$

$$= \sum_{i=1}^{n} \bar{\mu}_{T_s^{-1}\theta_i}(G)p(\theta_i)$$

$$= \mu(G)$$

is a stationary and ergodic decomposition of μ. But since μ is already ergodic,

$$\mu = \bar{\mu}_{T_s^{-1}\theta}$$

which implies that

$$\bar{H}(\mu_{T_s^{-1}\theta}) = \bar{H}(\mu). \tag{3}$$

Substituting (3) into (2) yields

$$\bar{H}(\nu_\theta) = \bar{L}_s(\theta)\bar{H}(\mu). \tag{4}$$

Now the code actually places Z into the FL-VL codes T_c. For each ergodic a.m.s. measure ν_θ, it follows as before that

$$\bar{H}(\sigma_\theta) = \frac{\bar{H}(\nu_\theta)}{\bar{L}_c(\theta)} \tag{5}$$

where $\bar{L}_c(\theta)$ is the average code length through T_c. Combining (4) and (5) yields

$$\bar{H}(\sigma_\theta) = \frac{\bar{L}_s(\theta)}{\bar{L}_c(\theta)} \bar{H}(\mu)$$

and thus

$$\bar{H}(\theta) = \sum_{i=1}^{n} \bar{H}(\sigma_{\theta_i})p(\theta_i)$$

$$= \left[\sum_{i=1}^{n} \frac{L_s(\theta_i)}{L_c(\theta_i)} p(\theta_i)\right] \bar{H}(\mu),$$

which completes the proof of the theorem.

REFERENCES

[1] G. Katona and G. Tusnady, "The principle of conservation of entropy in a noiseless channel," Studia Sci. Math. Hungar., Vol. 2, pp. 29-35, 1967.

[2] M. Jacobsen, "A new proof of the theorem of G. Katona and G. Tusnady," Studia Sci. Math. Hungar., Vol. 2, pp. 399-402, 1967.

[3] V. M. Sidel'nikov, "On statistical properties of transformations induced by finite automata," Kibernetika (Kiev) Vol. 6, pp. 1-14, 1965 (In Russian).

[4] I. Csiszar, G. Katona and G. Tusnady, "Information sources with different cost scales and the principle of conservation of entropy," Z. Wahrsch. und Verw. Gebiete Vol. 12, pp. 185-122. 1969.

[5] F. Jelinek and K. S. Schneider, "On variable-length-to-block coding," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 765-774, Nov. 1972.

[6] D. L. Cohn and L. Mulder, "A formal structure for source codes," unpublished paper.

[7] J. C. Kieffer and J. G. Dunham, "A class of asymptotically mean stationary codes," submitted for publication.

[8] R. M. Grey and J. C. Kieffer, "Asympototically mean stationary measures," to appear in Annals of Probability.

THE DESIGN OF MULTIDIMENSIONAL QUANTIZERS USING PREQUANTIZATION

Kerry D. Rines and Neal C. Gallagher, Jr.
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

A novel approach to the design of multidimensional quantizers is presented. This technique is used to design optimum uniform multidimensional quantizers that can be operated in real time. The quantizers are easily implemented using zero memory nonlinearities, linear transformations and univariate uniform step size quantizers.

I. INTRODUCTION

There is considerable interest in the use of multidimensional quantizers for the encoding of analog sources. Much of this interest has been generated from a theoretical standpoint. The multivariate quantization results of Zador [1] point to the advantages of multidimensional quantizers over univariate quantizers at high bit rates. Simply stated, the results indicate that the optimum per sample distortion decreases as the dimension of the quantizer increases. Therefore the potential exists to improve the performance of digital encoders by replacing univariate quantizers with multidimensional quantizers.

Recently the design of optimum multidimensional quantizers has been addressed. Computer algorithms for designing optimum quantizers of two or more dimensions have been presented by many authors, such as Linde et al [2]. The optimum quantizers are implemented using a search procedure to choose, from a specified output set, the output that is the smallest distance from the input. This implementation of the optimum quantizer may be difficult or impossible to operate in real time at high bit rates. In contrast the univariate uniform step size quantizer is a zero memory device that can be operated in real time. To date the easy implementation and real time operation of the univariate uniform step size quantizer has outweighed the theoretical advantages of using multidimensional quantizers in the design of digital encoders.

In this paper we present a novel approach to the design of multidimensional quantizers called prequantization. The design is illustrated in Figure 1 where a zero memory nonlinearity called a prequantizer precedes a specified multidimensional quantizer.

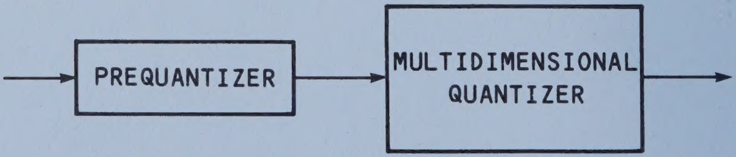


Figure 1. Multidimensional Quantizer Design using Prequantization.

This design is similar in some respects to the companding design of nonuniform univariate quantizers first proposed by Bennett [3]. In the univariate case a nonuniform quantizer may be difficult to implement directly. However, with companding we can design a nonuniform quantizer using a uniform step size quantizer, an invertible nonlinearity and the inverse nonlinearity. Similarly, prequantization can be used to design many multidimensional quantizers. Prequantization enables us to design these quantizers using a simple multidimensional quantizer, which is easy to implement and operate in real time, along with a zero memory nonlinearity. We illustrate the usefulness of prequantization with three examples.

In a recent paper Gersho [4] considers the partitioning of optimum uniform multidimensional quantizers. He states that the optimum uniform two-dimensional quantizer is the hexagonal quantizer. In three dimensions, Gersho argues that the truncated octahedral quantizer is very likely to be the optimum uniform three-dimensional quantizer. The analog of the truncated octahedron is considered for four dimensions. The resulting quantizer is not known to be optimal for four dimensions, but does have a lower per sample distortion than the three dimensional truncated octahedral quantizer. In this paper we present the designs for these three quantizers using prequantization. In each case the design is easy to implement and the quantizer can operate in real time. The real time operation of these quantizers for high bit rates is a significant result and demonstrates the important practical applications for prequantization. We begin in section II with a discussion of the prequantization design procedure.

II. PREQUANTIZATION

The design of k-dimensional quantizers using prequantization is illustrated in Figure 1. The design consists of a nonlinearity called a prequantizer preceding a specified k-dimensional quantizer. The implementation of this design approach takes place in two steps. First a k-dimensional quantizer meeting a specified criterion is chosen. In this paper we are interested in real time operation, therefore we specify that the quantizer be able to operate in real time. Examining Figure 1, we require that the real time (specified) quantizer have the same set of output values as the quantizer we wish to design. This is the only constraint placed on the choice of the real time quantizer. Free to choose from all quantizers satisfying the output constraint, we choose a real time quantizer that is easy to implement. The ability to exercise some control over the choice of the k-dimensional quantizer is one of the advantages of this design procedure.

The second step in the implementation is the design of the prequantizer. The role of the prequantizer is to complete the mapping of the input variables into the desired output values. The real time k-dimensional quantizer can be characterized by the mapping of its input space into its output values. This mapping is usually described by a partitioning of the input space, where all the input vectors contained within one partition are mapped into the same output vector. Since the real time quantizer is chosen based only on its output values, we do not expect its partitioning to be the same as the partitioning of the quantizer being designed. It is the prequantizer which is used to obtain the partitioning specified by the desired quantizer design. The prequantizing function maps a partition specified by the quantizer being designed into a partition of the real time quantizer that corresponds to the specified output. Once the prequantizing function is determined the k-dimensional quantizer design is complete. We illustrate the design procedure with a simple example.

Consider the design of a univariate quantizer with input x and output $\hat{x}$ as described in (1).

$$\tilde{x} = n\Delta \ ; \quad n\Delta - \frac{\Delta}{4} \le x < n\Delta + \frac{3\Delta}{4}. \tag{1}$$

Using the prequantization procedure, we first choose a quantizer that is easy to implement and has the same output set as given in (1). We choose the uniform step size quantizer given by

$$\hat{y} = n\Delta \ ; \quad n\Delta - \frac{\Delta}{2} \le y < n\Delta + \frac{\Delta}{2}. \tag{2}$$

We now determine the prequantizing function that must precede the quantizer in (2) to complete the design. Observe that quantizing $y = x - \frac{\Delta}{4}$ in (2) is identical to quantizing x in (1). Thus the prequantizing function is simply $f(x) = x - \frac{\Delta}{4}$ and the design of the quantizer in (1) is complete.

III. HEXAGONAL QUANTIZATION

Gersho has argued that the optimum uniform two-dimensional quantizer is the hexagonal quantizer. The design of a hexagonal quantizer using prequantizing is given here. First we attempt to find a two-dimensional quantizer that can be easily implemented and has the same set of output values as the hexagonal quantizer. One quantizer meeting these requirements is a scaled version of the diamond quantizer given below.

Let the inputs to the two-dimensional quantizer be x and y. The variables x and y are first encoded into two new variables w and z by the linear transformation,

$$\begin{aligned} w &= x + \sqrt{3}\, y \\ z &= x - \sqrt{3}\, y. \end{aligned} \tag{3}$$

The variables w and z are quantized separately by univariate quantizers with a uniform step size Δ. The outputs of the two-dimensional quantizer are then obtained using the linear transformation,

$$\begin{aligned} \hat{x} &= \frac{1}{2}(\hat{w} + \hat{z}) \\ \hat{y} &= \frac{1}{2\sqrt{3}}(\hat{w} - \hat{z}). \end{aligned} \tag{4}$$

The position of this quantizer in the hexagonal quantizer design is shown in Figure 2 and the partitioning of the scaled diamond quantizer is given in Figure 3. Having chosen the two-dimensional quantizer given in (3) and (4) we now turn to the design of the prequantizer.

The prequantizer must map the hexagonal region corresponding to each output into the scaled diamond shaped region corresponding to that same output. Consider the hexagonal partitioning shown in Figure 4.

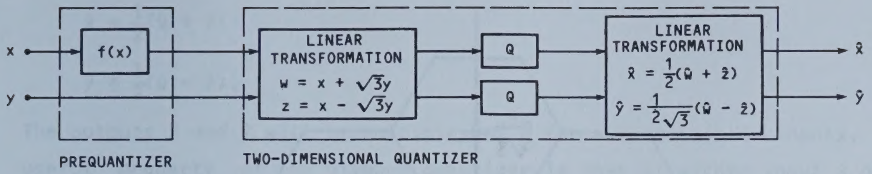

x
y
f(x)
LINEAR TRANSFORMATION
w = x + √3y
z = x − √3y
Q
Q
LINEAR TRANSFORMATION
x̂ = 1/2(ŵ + ẑ)
ŷ = 1/(2√3)(ŵ − ẑ)
x̂
ŷ
PREQUANTIZER
TWO-DIMENSIONAL QUANTIZER

Figure 2. Prequantization design for the hexagonal quantizer. The quantizer Q has uniform step-size Δ.

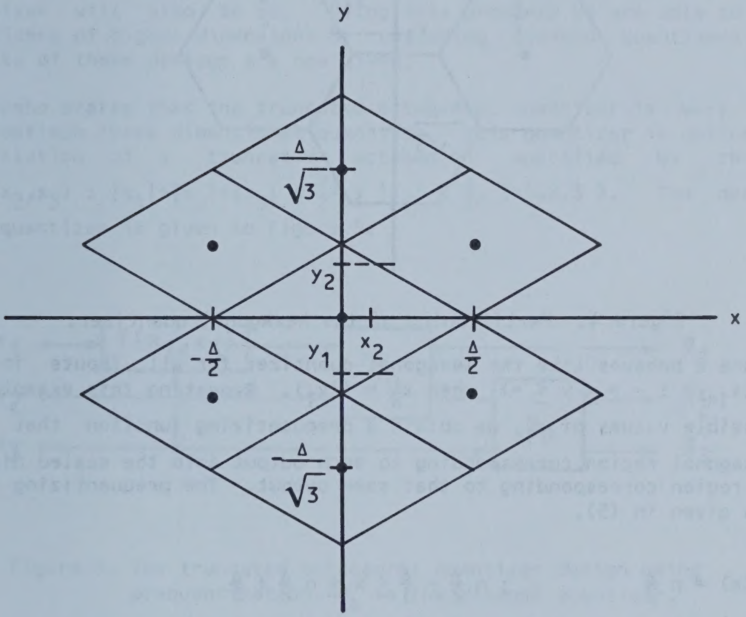

y
x
Δ/√3
y2
y1
x2
−Δ/2
Δ/2
−Δ/√3

Figure 3. Partitioning of the scaled diamond quantizer.

Assume x is fixed and the pair (x,y) is contained within a given hexagonal partition. We now pose the question, does there exist a value x' such that the pair (x',y) is contained within the corresponding diamond partition for all values of y? This approach is illustrated with the following example. Let $x = x_1$ as shown in Figure 3 and let y be in the range $-\frac{\Delta}{2\sqrt{3}}$ to $\frac{\Delta}{2\sqrt{3}}$. In Figure 4 we observe that the hexagonal quantizer output will be (0,0) for all input pairs in the set $\{(x_1,y) : y_1 \leq y \leq y_2\}$. Similarly in Figure 3 we observe that the scaled diamond quantizer output will be (0,0) for all input pairs in the set $\{(x_2, y) : y_1 \leq y \leq y_2\}$. Therefore if $x_2 = f(x_1)$, the quantizer in Figure 2 will behave like the hexagonal quantizer for all input pairs in the set $\{(x_1,y) : \frac{-\Delta}{2\sqrt{3}} \leq y \leq \frac{\Delta}{2\sqrt{3}}\}$. In fact, we can show that the quantizer

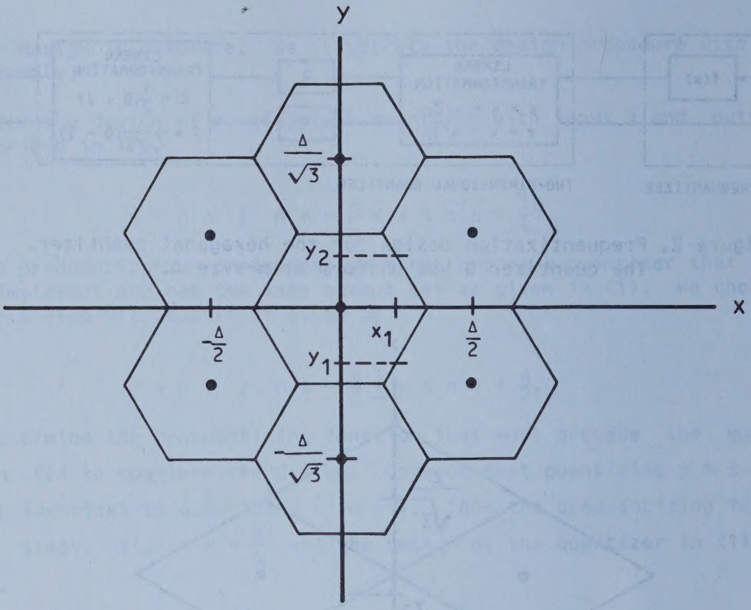


Figure 4. Partitioning of the hexagonal quantizer.

in Figure 2 behaves like the hexagonal quantizer for all inputs in the set $\{(x_1,y) : -\infty \leq y \leq \infty\}$ when $x_2 = f(x_1)$. Repeating this example for all possible values of x_1, we obtain a prequantizing function that maps the hexagonal region corresponding to each output into the scaled diamond shaped region corresponding to that same output. The prequantizing function is given in (5).

$$\begin{aligned} f(x) &= n\,\frac{\Delta}{2} &&;\ n\,\frac{\Delta}{2} - \frac{\Delta}{3} \leq x \leq n\,\frac{\Delta}{2} + \frac{\Delta}{3} \\ &= 3x - (2n+1)\,\frac{\Delta}{2} &&;\ n\,\frac{\Delta}{2} + \frac{\Delta}{3} \leq x \leq (n+1)\,\frac{\Delta}{2} - \frac{\Delta}{3}. \end{aligned} \tag{5}$$

IV. RESULTS IN HIGHER DIMENSIONS

In this section we present the design of the optimum (or near optimum) uniform quantizers for three and four dimensions. Each of these quantizers use in their designs a two-dimensional quantizer termed the diamond quantizer. The algorithm for the diamond quantizer is as follows. Let the inputs to the two-dimensional quantizer be x and y. The variables x and y are first encoded into two new variables w and z by the linear transformation,

$$\begin{aligned} w &= x + y \\ z &= x - y. \end{aligned} \tag{6}$$

The variables w and z are quantized separately by univariate quantizers with a uniform step size Δ. The outputs of the diamond quantizer are then obtained from a linear transformation of the quantized variables $\hat{w}$ and $\hat{z}$ given by

$$\hat{x} = \frac{1}{2}(\hat{w} + \hat{z})$$
$$\hat{y} = \frac{1}{2}(\hat{w} - \hat{z}). \tag{7}$$

The outputs $\hat{x}$ and $\hat{y}$ will be multiples of $\frac{\Delta}{2}$ for all possible inputs. A useful property of the diamond quantizer is that if either input x or y is a multiple of $\frac{\Delta}{2}$, its quantized value $\hat{x}$ or $\hat{y}$ will be that same multiple of $\frac{\Delta}{2}$. Therefore if the output of one diamond quantizer $\hat{x}$ is used as the input to a second diamond quantizer, the output of the second diamond quantizer will also be $\hat{x}$. Using this property we are able to design quantizers of higher dimensions by cascading diamond quantizers. The results of these designs are now given.

Gersho states that the truncated octahedral quantizer is very likely the optimum three dimensional quantizer. This quantizer is defined by a tessellation of a truncated octahedron specified by the set $\{(x_1,x_2,x_3) : |x_1|+|x_2|+|x_3| < \frac{3\Delta}{2} ;\ |x_i| < \frac{\Delta}{2},\ i=1,2,3\}$. The design of this quantizer is given in Figure 5.

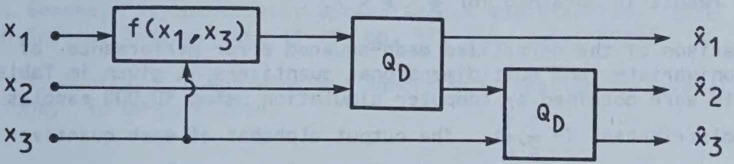


Figure 5. The truncated octahedral quantizer design using prequantization. Q_D is the diamond quantizer.

The prequantizing function is given in (8) where $e = |x_3| \bmod(0,\frac{\Delta}{2})$. For $e \leq \frac{\Delta}{4}$,

$$\begin{aligned} f(x_1,x_3) &= n\frac{\Delta}{2} &&;\ n\frac{\Delta}{2} - \frac{\Delta}{4} + e \leq x_1 \leq n\frac{\Delta}{2} + \frac{\Delta}{4} - e \\ &= x_1 - \frac{\Delta}{4} + e &&;\ n\frac{\Delta}{2} + \frac{\Delta}{4} - e \leq x_1 \leq (n+1)\frac{\Delta}{2} \\ &= x_1 + \frac{\Delta}{4} - e &&;\ (n-1)\frac{\Delta}{2} \leq x_1 \leq n\frac{\Delta}{2} - \frac{\Delta}{4} + e. \end{aligned} \tag{8}$$

A similar result is obtained for $\frac{\Delta}{4} \leq e \leq \frac{\Delta}{2}$.

The four dimensional analog of the truncated octahedral quantizer is defined by the tessellation of the polytope specified by the set $\{(x_1,x_2,x_3,x_4) : |x_1|+|x_2|+|x_3|+|x_4| < 2\Delta ;\ |x_i| \leq \frac{\Delta}{2},\ i=1,2,3,4\}$. For convenience we will call this quantizer the 4-d uniform quantizer. The design of the 4-d uniform quantizer is shown in Figure 6.

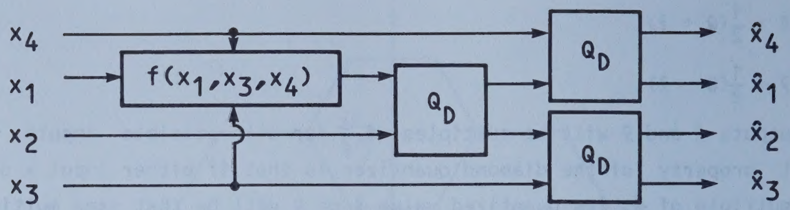


Figure 6. The 4-d uniform quantizer using prequantization. Q_D is the diamond quantizer.

The prequantizing function is given in (9) where $z = |x_3| \bmod(0,\frac{\Delta}{2})$, $w = |x_4| \bmod(0,\frac{\Delta}{2})$ and $e = z+w$. For $e \le \frac{\Delta}{2}$,

$$\begin{aligned} f(x_1,x_3,x_4) &= n\,\frac{\Delta}{2} \quad ; \ (n-1)\,\frac{\Delta}{2} + e \le x_1 \le (n+1)\frac{\Delta}{2} - e \qquad (9)\\ &= x_1 - \frac{\Delta}{2} + e \ ; \ (n+1)\,\frac{\Delta}{2} - e \le x_1 \le (n+1)\,\frac{\Delta}{2}\\ &= x_1 + \frac{\Delta}{2} - e \ ; \ (n-1)\,\frac{\Delta}{2} \le x_1 \le (n-1)\,\frac{\Delta}{2} + e. \end{aligned}$$

A similar result is obtained for $\frac{\Delta}{2} \le e \le \Delta$.

A comparison of the normalized mean-squared error performance of the uniform univariate and multidimensional quantizers is given in Table 1. The results were obtained by computer simulation using 30,000 samples uniformly distributed $(-\frac{1}{2},\frac{1}{2})$. The output alphabet of each quantizer was assigned one hundred quantization levels per input sample.

Dimension	Quantizer	nmse ($\times 10^{-5}$)
1	uniform step-size	9.99
2	hexagonal	9.66
3	truncated octahedral	9.48
4	4-d uniform	9.17

V. DISCUSSION

In this paper we have presented a new approach to the design of multidimensional quantizers. The usefulness of the prequantization approach has been demonstrated by the design of three optimum (or near optimum) uniform multidimensional quantizers. In each example the quantizer can be implemented using a zero memory nonlinearity, linear transformations, and univariate uniform step-size quantizers. As a result the computation time of each quantizer is independent of the output alphabet size. Therefore, these quantizers are both easy to implement and are able to operate in real time even at very high bit rates.

The prequantization design approach is also compatible with the design of nonuniform multidimensional quantizers. In [4] Gersho generalizes the companding technique for the design of nonuniform univariate quantizers to the design of nonuniform multidimensional quantizers. Bucklew [5] shows that an optimum k-dimensional quantizer can be designed using an optimum uniform k-dimensional quantizer, which is preceded by a multivariate invertible nonlinearity and followed by the inverse nonlinearity. Therefore the nonlinear prequantizing function used in optimum uniform k-dimensional quantizers is compatible and may even be of an advantage when the companding approach is applied to multidimensional quantizers.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support of the Air Force Office of Scientific Research under grant AFOSR 78-3605.

REFERENCES

[1] P. Zador, Development and Evaluation of Procedures for Quantizing Multivariate Distributions, Ph.D. Dissertation, Stanford University, 1964, University Microfilm No. 64-9855.

[2] Y. Linde, A. Buzo, R.M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Comm., Vol. COM-28, pp.84-95, January 1980.

[3] W.R. Bennett, "Spectra of Quantized Signals," B.S.T.J., Vol. 27, pp. 446-472, July, 1948.

[4] A. Gersho, "Asymptotically Optimal Block Quantization," IEEE Trans. on Inform. Theory, Vol. IT-25, pp. 373-380, July, 1979.

[5] J. A. Bucklew, "Companding and Random Quantization in Several Dimensions," to be published.

A NOTE ON MULTIDIMENSIONAL ASYMPTOTIC QUANTIZATION THEORY

J. A. BUCKLEW
Electrical and Computer Engineering Department
University of Wisconsin
Madison, Wisconsin 53705

G. L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

A rigorous development of the asymptotic theory of k dimensional quantization for r-th power distortion measures is presented. Subject only to a moment condition, it is shown that the infimum over all N level quantizers of the quantity $N^{r/k}$ times the r-th power distortion measure converges to a finite constant as $N \to \infty$.

I. INTRODUCTION

Several investigators [1-10] have considered the problem of designing a quantizer to minimize some given distortion measure. An important aspect of a quantizer is the distortion it will introduce into the data. In general, except for some rather trivial input data distributions, it is exceedingly difficult to predict the distortion performance of a quantizer until it is actually designed, and the design of even a relatively simple one dimensional quantizer can be difficult and time consuming. Hence an asymptotic theory is sometimes used to predict the distortion performance of quantizers. In this paper we present a rigorous formulation of the asymptotic theory of optimal multidimensional quantization subject to an r-th power distortion measure. The results contained herein are more general than previous efforts for this distortion measure.

II. DEVELOPMENT

In the sequel, r will be a fixed positive number, k will be a fixed positive integer, $\|\cdot\|$ will denote the L_2 norm on $\mathbb{R}^k$, λ will denote Lebesgue measure on $\mathbb{R}^k$, and X will denote a random variable taking values in $\mathbb{R}^k$ with distribution function F. Let b be a positive number. For any measurable function $h:\mathbb{R}^k \to [0,\infty)$, let $\|h\|_b$ denote the integral of h^b raised to the 1/b power, where all integrals, unless specified otherwise, are taken over $\mathbb{R}^k$. For an element x of $\mathbb{R}^k$, x_i will denote the i-th component of x. Let F_{ac} and F_s denote, respectively, the absolutely continuous and the singular components (with respect to λ) of F. The derivative of F_{ac} will be denoted by f. We will interpret 1/0 as ∞ and $0\cdot\infty$ as 0. We will let $U = \overset{k}{\underset{i=1}{X}} [0,1]$ denote the closed unit cube. The function $I_g: \mathbb{R} \to \mathbb{R}$ will denote the greatest integer function. We will specify a quantizer by specifying its output levels. The outputs are dictated by nearest neighbor assignment, and ties are broken in any arbitrary prescribed manner. In the sequel we will adopt the convention that distributing

N points uniformly over a cube refers to placing at least one point on each of the $[I_g(N^{1/k})]^k$ uniformly spaced points on the cube with arbitrary assignment of the remaining points in the cube.

Let $\mathcal{Q}_N$ denote the class of all N level k dimensional quantizers. Let

$$B_r(N) = \inf_{Q\in\mathcal{Q}_N} N^{r/k} \int \| x-Q(x) \|^r \, dF(x) \quad .$$

In the case where F has compact support, it is straightforward to show that an N level quantizer exists which will realize the infimum in the expression for $B_r(N)$. In this case, Q_N^o will denote any one of the quantizers which achieve the infimum.

The following result is due to Zador [7] who attributes the idea to J. M. Hammersley.

Proposition 1: Suppose X is uniformly distributed over U. Then the following limit exists: $\lim_{N\to\infty} B_r(N) = J_{k,r}$.

Proof: Let N and M be integers such that N>M>0, and for each such N and M define n by $Mn^k \leq N < M(n+1)^k$. Let $I_i = \left[\frac{i}{n+1}, \frac{i+1}{n+1}\right]$, i = 0,...,n, and consider the $(n+1)^k$ distinct cubes in $\mathbb{R}^k$ formed by all possible k dimensional products of the $\{I_i\}$. Pick n^k of the cubes, $C(1),\ldots,C(n^k)$, such that the remaining $(n+1)^k - n^k$ ones, whose union we will call V, are on the boundary of U. Note that $\lambda(V) = 1 - [n/(n+1)]^k$. Place M points in each of the cubes $C(1),\ldots,C(n^k)$ so as to minimize $B_r(N)$, and call the resulting quantizer Q_N^s. Notice that the distance between any $x\in V$ and the nearest representation point is upper bounded by $2\sqrt{k}/(n+1)$. Hence

$$\begin{aligned} N^{-r/k}B_r(N) &\leq \sum_{j=1}^{n^k} \int_{C(j)} \| x-Q_N^s(x) \|^r dx + \int_V \| x-Q_N^s(x) \|^r dx \\ &\leq n^k(n+1)^{-k-r}B_r(M)\, M^{-r/k} + \lambda(V)[2\sqrt{k}/(n+1)]^r . \end{aligned}$$

Thus $B_r(N) \leq [n/(n+1)]^k B_r(M) + (2\sqrt{k})^r M^{r/k}\lambda(V)$. Taking $\overline{\lim}$ in N and then taking $\underline{\lim}$ in M yields the desired result. QED

Remark: A simple change of variable yields that if X is uniformly distributed over a cube C then

$$\lim_{N\to\infty} B_r(N) = [\lambda(C)]^{r/k} J_{k,r} \; .$$

Remark: Gersho [9] calculated bounds for $J_{k,r}$ for various k and r. For example, $J_{1,2} = 1/12$.

Proposition 2: Consider p disjoint cubes $C_1,\ldots,C_p$ such that $\lambda(C_i) = m < \infty$ for i=1,...,p. Suppose X has a density given by $f(x)=p_i$ if $x\in C_i$, and $f(x)=0$ otherwise. Then $\lim_{N\to\infty} B_r(N) = J_{k,r} \| f \|_{k/(k+r)}$.

Proof: Let $L = \underline{\lim}_{N\to\infty} B_r(N)$. Let $N_i(N)$ represent the number of output levels of Q_N^o inside C_i, $i=1,\dots,p$, and let $N_0(N)$ be the number of output levels of Q_N^o outside $\bigcup_{i=1}^{p} C_i$. Define $r_i(N) = N_i(N)/N$. Let N' denote a subsequence of the positive integers such that $\lim_{N'\to\infty} B_r(N') = L$ and $\lim_{N'\to\infty} r_i(N) = R_i$ exists for $i=0,1,\dots,p$. Pick $\varepsilon>0$ and define $C_{i\varepsilon}$ as a parallel cube having the same center as C_i but with sides of length $m^{1/k}-\varepsilon$. Partition $C_i - C_{i\varepsilon}$ into M_ε cubes of side length ε, and put an output level into the center of each of these. Define $Q_{N'}^s$ to be the quantizer that has the same output levels as $Q_{N'}^o$ plus the pM_ε output levels distributed in the boundary regions. Then

$$B_r(N') \geq (N')^{r/k} \sum_{i=1}^{p} p_i \int_{C_{i\varepsilon}} \| x-Q_{N'}^s(x) \|^r dx .$$

Hence a straightforward argument yields

$$L \geq J_{k,r} \sum_{i=1}^{p} (p_i/R_i^{r/k}) [\lambda(C_{i\varepsilon})]^{1+r/k} .$$

Since $\varepsilon>0$ is arbitrary, we have

$$L \geq J_{k,r} \sum_{i=1}^{p} (p_i/R_i^{r/k}) m^{1+r/k} .$$

Minimizing the right hand side over all R_i under the constraint that $R_i \geq 0$ and $\sum_{i=1}^{p} R_i \leq 1$ yields $L \geq J_{k,r} \| f \|_{k/(k+r)}$.

Now construct another suboptimal quantizer $\tilde{Q}_N^s$ by putting N_i points optimally into C_i, $i=1,\dots,p$ where

$$N_i = I_g \left[Np_i^{k/(k+r)} \Big/ \sum_{j=1}^{p} p_j^{k/(k+r)} \right]. \tag{1}$$

It easily follows that $\lim_{N\to\infty} N_i/N = p_i^{k/(k+r)} \Big/ \sum_{j=1}^{p} p_j^{k/(k+r)}$.

Hence $B_r(N) \leq N^{r/k} \sum_{i=1}^{p} \int_{C_i} p_i \| x-\tilde{Q}_N^s(x) \|^r dx$, and it follows that

$$\overline{\lim}_{N\to\infty} B_r(N) \leq J_{k,r} \left[\sum_{i=1}^{p} p_i^{k/(k+r)} m \right]^{r/k+1} = J_{k,r} \| f \|_{k/(k+r)} .$$

QED

Proposition 3: Let S be a cube with $\lambda(S) < \infty$, and assume that X is uniformly distributed on S. Let C be a finite union of disjoint cubes having the same Lebesgue measure and contained in S. Then

$$\lim_{N\to\infty} [N^{r/k}/\lambda(S)] \int_C \| x-Q_N^o(x) \|^r dx = J_{k,r} \frac{\lambda(C)[\lambda(S)]^{r/k}}{\lambda(S)} .$$

Proof: Assume that $\lambda(C)$ does not equal zero or $\lambda(S)$. Let $N_C(N)$ be the number of output levels of Q_N^0 inside C. Let N' be a subsequence such that

$$\overline{\lim_{N\to\infty}} \int_C \frac{N^{r/k}}{\lambda(S)} \|x-Q_N^0(x)\|^r dx = \lim_{N'\to\infty} \int_C \frac{(N')^{r/k}}{\lambda(S)} \|x-Q_{N'}^0(x)\|^r dx$$

and $r_C(N') = N_C(N')/N' \to R_C$ as $N'\to\infty$. Fix ε and M_ε as in the proof to Proposition 2 but now for the set C and its subset C_ε. A straightforward development yields that

$$\lim_{N'\to\infty} (N')^{r/k} \int_C \frac{\|x-Q_{N'}^0(x)\|^r}{\lambda(S)} dx \geq \frac{J_{k,r}[\lambda(C)]^{1+r/k}}{R_C^{r/k}\,\lambda(S)} .$$

Suppose the above is met with a strict inequality. We know that

$$\lim_{N'\to\infty} (N')^{r/k} \int_S \frac{\|x-Q_{N'}^0(x)\|^r}{\lambda(S)} dx = J_{k,r}[\lambda(S)]^{r/k}$$

$$= \min_{R\in[0,1]} \left[\frac{[\lambda(C)]^{1+r/k} J_{k,r}}{\lambda(S)\, R^{r/k}} + \frac{[\lambda(S-C)]^{1+r/k} J_{k,r}}{(1-R)^{r/k}\,\lambda(S)} \right].$$

Hence, $$\lim_{N'\to\infty} (N')^{r/k} \int_{S-C} \frac{\|x-Q_{N'}^0(x)\|^r}{\lambda(S)} dx < \frac{[\lambda(S-C)]^{1+r/k} J_{k,r}}{(1-R_C)^{r/k}\,\lambda(S)} .$$

There are no more than $(1-r_C(N'))/N'$ output levels of $Q_{N'}^0$ inside S-C. A standard ε construction yields that

$$\lim_{N'\to\infty} (N')^{r/k} \int_{S-C} \frac{\|x-Q_{N'}^0(x)\|^r}{\lambda(S)} dx \geq \frac{[\lambda(S-C)]^{1+r/k} J_{k,r}}{\lambda(S)(1-R_C)^{r/k}} ,$$

which implies a contradiction. Hence, the above implies

$$[\lambda(S)]^{r/k+1} J_{k,r} = \lambda(S) \lim_{N'\to\infty} B_r(N')$$

$$= \lim_{N'\to\infty} (N')^{r/k} \int_C \|x-Q_{N'}^0(x)\|^r dx + \lim_{N'\to\infty} (N')^{r/k} \int_{S-C} \|x-Q_{N'}^0(x)\|^r dx$$

$$= [\lambda(C)]^{1+r/k} J_{k,r}/R_C^{r/k} + [\lambda(S-C)]^{1+r/k} J_{k,r}/(1-R_C)^{r/k}$$

and $R_C = \lambda(C)/\lambda(S)$.

Suppose $\overline{\lim_{N\to\infty}} N^{r/k} \int_{S-C} \frac{\|x-Q_N^0(x)\|^r}{\lambda(S)} dx = B > \frac{\lambda(S-C)[\lambda(S)]^{r/k} J_{k,r}}{\lambda(S)}$. Let N" be a subsequence such that $\lim_{N''\to\infty} (N'')^{r/k} \int_{S-C} \frac{\|x-Q_N^0(x)\|^r}{\lambda(S)} dx = B$. Since $\lim_{N''\to\infty} B_r(N'') = [\lambda(S)]^{r/k} J_{k,r}$, we therefore have

$$\lim_{N''\to\infty} (N'')^{r/k} \int_C \frac{\|x-Q^o_{N''}(x)\|^r}{\lambda(S)}\, dx < \frac{\lambda(C)[\lambda(S)]^{r/k}\, J_{k,r}}{\lambda(S)}$$

which implies a contradiction because the $\underline{\lim}$ of the sequence on the left hand side was equal to the right hand side. Therefore,

$$\overline{\lim_{N\to\infty}}\, N^{r/k} \int_{S-C} \frac{\|x-Q^o_N(x)\|^r}{\lambda(S)}\, dx \le \frac{\lambda(S-C)[\lambda(S)]^{r/k}\, J_{k,r}}{\lambda(S)} \quad \text{and}$$

$$\underline{\lim_{N\to\infty}}\, N^{r/k} \int_{C} \frac{\|x-Q^o_N(x)\|^r}{\lambda(S)}\, dx \ge \frac{\lambda(C)[\lambda(S)]^{r/k}\, J_{k,r}}{\lambda(S)} .$$

Since C and S-C are the same type of region (unions of cubes), we may repeat all of the above arguments switching the roles of S-C and C to obtain the proposition. If $\lambda(C)$ equals zero or $\lambda(S)$, the result is obvious. QED

Let C_i, $i=1,\dots,p$ be a collection of disjoint cubes of Lebesgue measure $m<\infty$. Partition each C_i into 2^{qk} (q is a positive integer) cubes C_{ij}, $j=1,\dots,2^{qk}$, $i=1,\dots,p$, such that $\lambda(C_{ij})$ is the same for all i,j. Define a probability density on the C_i's as $\overline{f}_p(x) = p_i$ if $x \in C_i$, $i=1,\dots,p$, and zero otherwise. Define a probability density on the C_{ij}'s as $f_{2^{qk}p}(x) = p_{ij}$ if $x \in C_{ij}$, $i=1,\dots,p$, $j=1,\dots,2^{qk}$, and zero otherwise. We add the further constraint that

$$\sum_{j=1}^{2^{qk}} p_{ij} = p_i 2^{qk}, \quad i=1,\dots,p.$$

Define a quantizer Q_N on the C_i's by putting N_i, given by (1), points optimally in the C_i-th cube. From the proof of Proposition 2 we know that if X has the density $\overline{f}_p$ then

$$\lim_{N\to\infty} N^{r/k} \int \|x-Q_N(x)\|^r\, \overline{f}_p(x)\,dx = J_{k,r}\, \|\overline{f}_p\|_{k/(k+r)} .$$

Hence, this quantizer is asymptotically optimum for $\overline{f}_p$.

Proposition 4: Given the above, we have

$$\lim_{N\to\infty} N^{r/k} \int \|x-Q_N(x)\|^r\, f_{2^{qk}p}(x)\,dx = J_{k,r}\, \|\overline{f}_p\|_{k/(k+r)} .$$

Proof: We give the following argument for the limit superior, but it is identical for the limit inferior with the corresponding inequalities reversed. Let Q^i_N denote the optimum N level quantizer for $f(x) = 1/m$ on C_i and zero otherwise. Then

$$\overline{\lim_{N\to\infty}}\, N^{r/k} \int \|x-Q_N(x)\|^r\, f_{2^{qk}p}(x)\,dx \le \sum_{i=1}^{p} \overline{\lim_{N\to\infty}}\, N^{r/k} \int_{C_i} \|x-Q_N(x)\|^r\, f_{2^{qk}p}(x)\,dx$$

$$= \sum_{i=1}^{p} \overline{\lim_{N\to\infty}}\, N^{r/k} \int_{C_i} \|x-Q^i_{N_i}(x)\|^r\, f_{2^{qk}p}(x)\,dx$$

$$\le \sum_{i=1}^{p} \sum_{j=1}^{2^{qk}} \overline{\lim_{N\to\infty}} \left(\frac{N}{N_i}\right)^{r/k} (N_i)^{r/k} \int_{C_{ij}} \|x-Q^i_{N_i}(x)\|^r\, p_{ij}\,dx .$$

Applying Proposition 3, we can upper bound the above by

$$\sum_{i=1}^{p}\sum_{j=1}^{2^{qk}} (\alpha_i)^{-r/k}\, p_{ij}\, J_{k,r}\, \lambda(C_{ij}) m^{r/k}, \text{ where } \alpha_i = p_i^{k/(k+r)} \Big/ \sum_{n=1}^{p} p_n^{k/(k+r)}. \qquad (2)$$

Since $\lambda(C_{ij})2^{qk} = m$, the above may be written as

$$J_{k,r}\left[\sum_{n=1}^{p} p_n^{k/(k+r)}\, m\right]^{1+k/r} = J_{k,r}\, \|\overline{f}_p\|_{k/(k+r)}. \qquad \text{QED}$$

Now assume that X has a density function f with compact support. Let a and b be finite and such that the support of f is contained in $C_f = \mathop{X}_{i=1}^{n} (a,b)$. Consider a partition of C_f into n^k cubes C_i such that $\lambda(C_i) = m$, $i=1,\dots,n^k$. Define a probability density f_n on C_f by $f_n(x) = p_j = P(X \in C_j)$ if $x \in C_j$ and is zero otherwise.

Proposition 5: Assuming the above, we have $B_r(N) \to J_{k,r}\, \|f\|_{k/(k+r)}$ as $N \to \infty$.

Proof: In [11, p.214] it is shown that $\lim_{n\to\infty} f_n(x) = f(x)$ a.e.$[\lambda]$. Define a quantizer Q_N^n for f_n by putting $I_g(\alpha_i N)$ output levels optimally inside C_i where α_i is given by (2) with $p = n^k$. From the proof of Proposition 2 we have

$$\lim_{N\to\infty} N^{r/k} \int \| x - Q_N^n(x)\|^r\, f_n(x)\, dx = J_{k,r}\, \|f_n\|_{k/(k+r)}\ ,$$

and from Proposition 4 we have $\lim_{N\to\infty} N^{r/k} \int \| x - Q_N^n(x)\|^r f_{2^q n}(x)\, dx = J_{k,r} \|f_n\|_{k/(k+r)}$.

Let $d \in (0,1)$ and define $Q_N^{n,d}$ to be a quantizer that has the same output levels as $Q^n_{I_g[(1-d)N]}$ plus $I_g(dN)$ output levels distributed uniformly on a rectangular grid over C_f. It is a simple matter to show $N^{r/k}\| x - Q_N^{n,d}(x)\|^r < K_{n,d}$ for $x \in C_f$ and where $K_{n,d}$ is a positive constant not depending on N or x. Notice that the discontinuities have F-measure zero. Hence by [12,p.31] and the fact that $\lim_{q\to\infty} f_{2^q n}(x) = f(x)$ a.e.$[\lambda]$ implies weak convergence via Scheffé's theorem [12,p.224], we find

$$\lim_{q\to\infty} N^{r/k} \int \| x - Q_N^{n,d}(x)\|^r f_{2^q n}(x)dx = N^{r/k} \int \| x - Q_N^{n,d}(x)\|^r f(x)dx\ . \qquad (3)$$

Now $\int N^{r/k} \| x - Q_N^{n,d}(x)\|^r\ |f_{2^q n}(x) - f(x)|dx \le K_{n,d} \int |f_{2^q n}(x) - f(x)|dx \to 0$ as $q \to \infty$ by the dominated convergence theorem. This implies that we have a uniform limit in N in (3). Therefore, $\overline{\lim}_{N\to\infty} N^{r/k} \int \| x - Q_N^{n,d}(x)\|^r\ f(x)dx$ $\le \overline{\lim}_{q\to\infty}\ \overline{\lim}_{N\to\infty} N^{r/k} \int \| x - Q_N^{n,d}(x)\|^r\ f_{2^q n}(x)dx$, where we have switched limits due to the uniform convergence in N. The above upper bound is further upper bounded by

$$\overline{\lim_{q\to\infty}}\ \overline{\lim_{N\to\infty}}\ N^{r/k} \int \| x - Q^n_{I_g[(1-d)N]}(x)\|^r\ f_{2^q n}(x)dx$$

$= J_{k,r}\,(1-d)^{-r/k}\,\| f_n \|_{k/(k+r)}$ via Proposition 4. Since d is arbitrary we have that there exists a sequence of quantizers $\{Q'_N\}$ such that

$$\overline{\lim_{N\to\infty}}\, N^{r/k} \int \| x-Q'_N(x) \|^{r}\, f(x)dx \le J_{k,r}\, \| f_n \|_{k/(k+r)} .$$

With Jensen's inequality we can show that $\| f_{2^q n} \|_{k/(k+r)}$ is monotone non-increasing in q. Using the same arguments for $f_{2^q n}$ as we did for f_n we have that there must exist a sequence of quantizers Q''_N such that $\overline{\lim_{N\to\infty}}\, N^{r/k} \int \| x-Q''_N(x) \|^{r} f(x)dx \le \lim_{q\to\infty} J_{k,r} \| f_{2^q n} \|_{k/(k+r)}$. It follows from Jensen's inequality that as $q\to\infty$, $f_{2^q n}(x) \to f(x)$ in $L_{k/(k+r)}$, which implies $\| f_{2^q n} \|_{k/(k+r)} \to \| f \|_{k/(k+r)}$ as $q\to\infty$. Suppose $\underline{\lim}_{N\to\infty} B_r(N) = L < J_{k,r} \| f \|_{k/(k+r)}$. Define $\delta = J_{k,r} \| f \|_{k/(k+r)} - L > 0$, and let $d \in (0,1)$. Let Q_N^d be the quantizer with the same output levels as $Q^{o}_{I_g[(1-d)N]}$ plus $I_g(dN)$ output levels distributed uniformly over a rectangular grid on C_f. Mimicking the arguments given above for the upper bound, we have

$$\lim_{q\to\infty}\ \underline{\lim}_{N\to\infty}\, N^{r/k} \int \| x-Q_N^d(x) \|^{r} f_{2^q n}(x)dx \le L\,(1-d)^{-r/k} .$$

Choose d such that $L < L\,(1-d) < L + \delta/2$. But we also have that $L + \delta/2 < J_{k,r} \| f \|_{k/(k+r)} \le J_{k,r} \| f_{2^q n} \|_{k/(k+r)}$ for $q = 0,1,2,\ldots$. Hence there exists a subsequence of the positive integers $\{q'\}$ and a number M such that for all $q' > M$,

$$\underline{\lim}_{N\to\infty}\, N^{r/k} \int \| x-Q_N^d(x) \|^{r} f_{2^{q'} n}(x)dx < L + \delta/2 < J_{k,r} \| f_{2^{q'} n} \|_{k/(k+r)}$$

which via Proposition 2 is a contradiction. QED

Now let F be a distribution function that is singular with respect to λ but still having compact support contained in C_f.

Proposition 6: Assuming the above, we have $\lim_{N\to\infty} B_r(N) = 0$.

Proof: For any $\varepsilon > 0$, there exists an open set S such that S has unit F-measure and $\lambda(S) < \varepsilon$. Since S is open, there exists a countable set of disjoint open cubes $\{I_i\}$ such that $S = \bigcup_{i=1}^{\infty} I_i$. Define Q_N by distributing N/2 output levels uniformly over C_f and by distributing $N_i = I_g\left[N\,\lambda(I_i) \Big/ \sum_{j=1}^{q} \lambda(I_j)\right]$ additional output levels uniformly over I_i, $i = 1,\ldots,q$, where q is a positive integer such that $2^{r/k}(b-a)^r \sum_{j=q+1}^{\infty} \mu(I_j) < \varepsilon^{r/k}$, where μ is the measure corresponding to F. Let $\alpha = \min_{1\le i\le q} \lambda(I_i) \Big/ \sum_{j=1}^{q} \lambda(I_j)$. For $N > 2^k/\alpha$, if $x \in \bigcup_{i=1}^{q} I_i$, then $\| x-Q_N(x) \| \le \varepsilon^{1/k}\sqrt{k} \Big/ (\min_{1\le i\le q} N_i^{1/k} - 1) \le \varepsilon^{1/k}\sqrt{k} \big/ [(N\alpha-1)^{1/k} - 1] \le \varepsilon^{1/k}\sqrt{k} \big/ [(N\alpha)^{1/k} - 2]$.

For $N > 2^{k+1}$, if $x \in C_f$, then $\| x-Q_N(x) \| \le (b-a)\sqrt{k} \,/\, [(N/2)^{1/k} - 1]$.

Therefore, $\overline{\lim\limits_{N\to\infty}}\, B_r(N) \le (\alpha\varepsilon)^{r/k} k^{r/2} + (b-a)^r k^{r/2} 2^{r/k} \sum\limits_{j=q+1}^{\infty} \mu(I_j) < 2(k\varepsilon)^{r/k}$. QED

Now let F be an arbitrary distribution function whose support lies inside C_f. Notice that we can write $F(x) = \alpha F_{ac}(x) + (1-\alpha)F_s(x)$ for some $\alpha \in [0,1]$.

Proposition 7: Assuming the above, we have $\lim\limits_{N\to\infty} B_r(N) = \alpha J_{k,r} \| f \|_{k/(k+r)}$.

Proof: Let $d \in (0,1)$, and define a quantizer Q_N^d such that $I_g[(1-d)N]$ output levels are distributed optimally for F_{ac} (denote this quantizer as $Q^{ac}_{I_g[(1-d)N]}$) and $I_g(dN)$ are distributed optimally for F_s (denote this as $Q^s_{I_g(dN)}$). Then $N^{r/k} \int \| x - Q_N^d(x) \|^r dF(x) = \alpha\, N^{r/k} \int \| x - Q^{ac}_{I_g[(1-d)N]}(x) \|^r dF_{ac}(x) + (1-\alpha) N^{r/k} \int \| x - Q^s_{I_g(dN)}(x) \|^r dF_s(x)$. Taking the limit in N and involving the arbitrariness of d yields the result. QED

Bennett [6] modeled a nonuniform (one dimensional) quantizer as a zero memory nonlinearity $G(\cdot)$ followed by an N level uniform quantizer Q_N in turn followed by the inverse nonlinearity $h(\cdot) \triangleq G^{-1}(\cdot)$. The output of such a system (sometimes called a compander) could be written as $G^{-1}[Q_N[G(x)]]$ where x is the input. We will assume that the input to the compander X is a scalar (k=1) random variable with a density function f. The quantizer Q_N will be an N level uniform quantizer on [0,1]. The function G will be an increasing differentiable invertible mapping from $\mathbb{R}$ into [0,1]. Furthermore, $G'(x) = g(x)$ will be assumed to be continuous a.e. $[\lambda]$ and $g(|x|)$ will be nonincreasing in x for $|x| > M > 0$. Also, we assume that the support of f is a subset of the support of g.

Theorem 1: Given the above, if there exists $\delta>0$ such that $E\{[g(X)]^{-r-\delta}\} <\infty$, then $\lim\limits_{N\to\infty} N^r E\{|X-\hat{X}|^r\} = (1/12)E\{[g(X)]^{-r}\}$, when X is the input to the compander and $\hat{X}$ is the output.

Proof: Let $\mathscr{B}$ be the class of Borel sets in [0,1]. Define measures μ_N and μ on $\mathscr{B}$ as

$$\mu_N(A) = \int_A N^r |x-Q_N(x)|^r p(x)dx \text{ and } \mu(A) = (1/12) \int_A p(x)dx \text{ for}$$

$A \in \mathscr{B}$ where $p(x) = f[h(x)]/g[h(x)]$. From a generalization of the Riemann-Lebesgue lemma [14,p.13], we have that $\mu_N(A) \to \mu(A)$ as $N\to\infty$ for all $A \in \mathscr{B}$.

Also, define $\bar{f}_N(x) = \left| \dfrac{h(x) - h[Q_N(x)]}{x - Q_N(x)} \right|^r$ and $\bar{f}(x) = |h'(x)|^r$. Then $\bar{f}_N(x) \to \bar{f}(x)$ a.e. $[\lambda]$ as $N\to\infty$. Suppose that for some $\varepsilon>0$, $\int (\bar{f}_N(x))^{1+\varepsilon} p(x)dx < \bar{M} < \infty$ for every N. By a straightforward truncation argument, application of the generalized dominated convergence theorem [15,p.232] yields $\lim\limits_{N\to\infty} \int \bar{f}_N(x)\, \mu_N(dx) \to \int \bar{f}(x)\, \mu(dx)$. Since $N^r E\{|X-\hat{X}|^r\} = \int \bar{f}_N(x)\, \mu_N(dx)$ and $\int \bar{f}(x)\, \mu(dx) = (1/12) \int |h'(x)|^r p(x)dx = 1/12\, E\{[g(X)]^{-r}\}$, the theorem would be proved. Therefore, define

$$g_N(x) = \begin{cases} \sup\limits_{t\epsilon\left[\frac{i}{N},\frac{i+1}{N}\right]} \bar{f}(t) = (g_i)^r & \text{for } x\epsilon \left[\frac{i}{N},\frac{i+1}{N}\right) , \; i = 1,\ldots,N-2 \\ \bar{f}(x) & \text{for } x\epsilon \left[0,\frac{1}{2N}\right) \cup \left[\frac{2N-1}{2N},1\right] \\ \bar{f}[Q_N(x)] & \text{for } x\epsilon \left[\frac{1}{2N},\frac{1}{N}\right] \cup \left[\frac{N-1}{N},\frac{2N-1}{2N}\right) \end{cases} .$$

Hence for N sufficiently large, because of the "tail" monotonicity of $g(x)$ and hence that of $h'(x)$ and because of the mean value theorem for derivatives, we have $g_N(x) \geq \bar{f}_N(x)$. Let $I = \int [g(x)]^{-r(1+\varepsilon)} \bar{f}(x)dx = \int_0^1 |h'(x)|^{r(1+\varepsilon)} p(x)\cdot dx$. Notice that by assumption $I<\infty$. Then since $h'(x)$ is continuous a.e.$[\lambda]$, we have $\int [g_N(x)]^{1+\varepsilon} p(x)dx \to I$ as $N\to\infty$, and the theorem is proved. QED

Remark: Suppose that the distribution of X is not absolutely continuous. Then we would like to consider

$$\overline{\lim_{N\to\infty}}\, N^r \int \left|\frac{h(x)-h[Q_N(x)]}{x-Q_N(x)}\right|^r |x-Q_N(x)|^r \, d\tilde{F}(x) \tag{5}$$

$\leq \overline{\lim\limits_{N\to\infty}}\, 2^{-r} \int \left|\frac{h(x)-h[Q_N(x)]}{x-Q_N(x)}\right|^r d\tilde{F}(x)$, where $\tilde{F}$ is the distribution function of $G(X)$. If $G(\cdot)$ has the same restrictions as in Theorem 1, then the left hand side of (5) exists as a limit and we have that it is upper bounded by $2^{-r} \int [g(x)]^{-r} dF(x)$.

Remark: Suppose for some $\varepsilon>0$ we let $g(x)$ be a probability density function such that $g(x)$ equals C for $|x| < x_o$ and equals $D|x|^{-(1+\varepsilon/2r)}$ for $|x| \geq x_o$. This $g(\cdot)$ satisfies all the assumptions of Theorem 1 and the above remark if $E\{|X|^{r+\varepsilon}\} < \infty$. Hence there exists a sequence of quantizers Q'_N such that $\overline{\lim\limits_{N\to\infty}}\, N^r \int |x-Q'_N(x)|^r dF(x) \leq (2D)^{-r} \int_{|x|\geq x_o} |x|^{r+\varepsilon/2} dF(x) + (2C)^{-r} \int_{|x|<x_o} dF(x)$.

This will prove to be a crucial point in the next theorem.

Now we return our attention to random variables taking values in $\mathbb{R}^k$. Let F be an arbitrary distribution function and express it as $F(x)=\alpha F_{ac}(x)+(1-\alpha)F_s(x)$.

Theorem 2: If $E\left\{\|X\|^{r+\varepsilon}\right\} < \infty$ for some $\varepsilon>o$, then $\lim\limits_{N\to\infty} B_r(N)=\alpha J_{k,r}\|f\|_{k/(k+r)}$.

Proof: Let C_t be a cube centered at the origin with its 2^k sides each of length t and parallel to the coordinate axes. Denote the marginal distributions as $F_i(x_i) = \alpha_i F_{i(ac)}(x_i) + (1-\alpha_i)F_{i(s)}(x_i)$. Since $|x_i|^{r+\varepsilon} \leq \|x\|^{r+\varepsilon}$, $E\left\{\|X\|^{r+\varepsilon}\right\} < \infty$ implies that $E\left\{|X_i|^{r+\varepsilon}\right\} < \infty$, $i = 1,\ldots,k$. Let Q_N denote the optimum N level quantizer for the conditional distribution $F(x|X\epsilon C_t)$. Let Q^i_N be a one dimensional quantizer for $F_i(x_i|\;|X_i| \geq t/2)$ such that $\overline{\lim\limits_{N\to\infty}}\, N^r \int |x-Q^i_N(x)|^r dF_i(x|\;|X_i| \geq t/2) \leq (2D)^{-r} \int |x|^r dF_i(x|\;|X_i| \geq t/2)$.

We may do this because of the second remark after Theorem 1. Let $d\epsilon(0,1)$. Then $\left[N^{r/k} \int_{C_t} \|x-Q_N(x)\|^r dF(x|\;X \epsilon C_t)\right] P(X \epsilon C_t) \leq B_r(N) \leq$

$$N^{r/k} \int_{C_t} \|x-Q_{I_g(dN)}(x)\|^r dF(x) + N^{r/k} \sum_{i=1}^{k} \int_{|x_i|\geq t/2} |x_i-Q^i_\alpha(x_i)|^r dF_i(x_i) \tag{6}$$

where $\alpha = I_g\left[(I_g[(1-d)N])^{1/k}\right]$. Taking the limit superior as N→∞, we find

$$\alpha\left[\int_{C_t} [f(x)]^{k/(k+r)}dx\right]^{(k+r)/k} \le \overline{\lim_{N\to\infty}}\, B_r(N) \le \frac{\alpha}{d^{r/k}}\left[\int_{C_t} [f(x)]^{k/(k+r)}dx\right]^{(k+r)/k}$$

$$+ \sum_{i=1}^{k} (1-d)^{-r/k}\,(2D)^{-r} \int_{|x_i|\ge t/2} |x_i|^{r+\varepsilon/2} dF_i(x_i).$$ Let t→∞ to yield that

$\alpha\| f\|_{k/(k+r)} \le \overline{\lim}\, B_r(N) \le \alpha\, d^{-r/k}\| f\|_{k/(k+r)}$. Since d is arbitrary, we get $\overline{\lim_{N\to\infty}}\, B_r(N) = \alpha\| f\|_{k/(k+r)}$. Trivially from the first inequality in (6)

$\underline{\lim_{N\to\infty}}\, B_r(N) \ge \alpha \left[\int_{C_t} [f(x)]^{k/(k+r)}dx\right]^{(k+r)/k}$ and taking the limit as t→∞ we get the desired result. QED

Remark: Utilizing Holder's inequality we have that if p(x) is a probability density function such that $\int \| x\|^{r+\delta} p(x)dx < \infty$ for some $\delta>0$, then $\| p\|_{k/(k+r)} < \infty$. Hence we know that the constant in the statement of Theorem 2 is finite.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research under Grant AFOSR-76-3062.

REFERENCES: [1] S.P. Lloyd, "Least squares quantization in PCM," unpublished memorandum, Bell Laboratories, 1957. [2] J. Max, "Quantizing for minimum distortion," IRE Trans. Info. Theory, IT-6, pp.7-12, March 1960. [3] Y. Linde, A. Buzo, and R.M. Gray, "An Algorithm for Vector Quantizer Design," IEEE Trans. Comm., Vol. COM-28, pp.84-95, Jan., 1980. [4] P.F. Panter and W. Dite, "Quantization in pulse-count modulation with nonuniform spacing of levels," Proc. IRE, Vol. 39, pp.44-48, 1951. [5] V.R. Algazi, "Useful approximations to optimum quantization," IEEE Trans. Commun. Tech., Vol. COM-14, pp.297-301, 1966. [6] W.R. Bennett, "Spectra of quantized signals," BSTJ, Vol. 27, pp.446-472, July 1948. [7] P. Zador, Development and Evaluation of Procedures for Quantizing Multivariate Distributions, Ph.D. Dissertation, Stanford University, 1964, University Microfilm No.64-9855. [8] P. Zador, "Asymptotic quantization of continuous random variables," unpublished memorandum, Bell Laboratories, 1966. [9] A. Gersho, "Asymptotically optimal block quantization," IEEE Trans. Info. Theory, Vol. IT-25, pp.373-380, July 1979. [10] Y. Yamada, S. Tazaki, and R.M. Gray, "Asymptotic performance of block quantizers with difference distortion measures," IEEE Trans. Info. Theory, Vol. IT-26, pp.6-14, Jan. 1980. [11] N. Dunford and J.T. Schwartz, Linear Operators - Part I, John Wiley and Sons, 1958. [12] P. Billingsley, Convergence of Probability Measures, John Wiley and Sons, 1968. [13] M. Loève, Probability Theory - I, 4th ed., Springer-Verlag, 1977. [14] I.P. Natanson, Theory of Functions of a Real Variable Vol. II, Ungar, 1960. [15] H.L. Royden, Real Analysis, MacMillan, 1968.

ON THE COMPLEXITY OF FIXED-PRIORITY SCHEDULING OF REAL-TIME TASKS*

J. Y-T LEUNG and J. WHITEHEAD
Northwestern University, Evanston, Illinois

ABSTRACT

We consider the complexity of determining whether a set of periodic, real-time tasks can be scheduled on $m \geq 1$ identical processors with respect to fixed-priority scheduling. It is shown that the problem is NP-hard in all but one special case. The complexity of optimal fixed-priority scheduling algorithms is also discussed.

INTRODUCTION

We consider the problem of scheduling a set of periodic, real—time tasks on both single-processor and multi-processor computing systems. Our interest in this particular scheduling problem is motivated by a situation in which computers are used to control real-time processes. In many of these real-time applications, the computer is required to execute programs in response to periodic, external signals and to guarantee that each such program be completely processed before the occurrence of a subsequent signal.

A periodic, real-time task T is characterized by the quadruple (s,e,d,p) with $0 < e \leq d \leq p$. In this characterization, task T initially makes a request for execution of a computation at time s, and thereafter at the times s + kp, (k = 1,2,...), with p being the period of T. The execution time required for each request of T is e time units, and a deadline for T occurs d time units after each request, by which time T must have completed its execution. A task system R consists of a set of n periodic, real-time tasks and is denoted by the quintuple $(\{T_i\},\{s_i\},\{e_i\},\{d_i\},\{p_i\})$, where each T_i $(1 \leq i \leq n)$ is a periodic, real-time task. Without loss of generality and for convenience in notation, we assume $\min_{1\leq i\leq n} \{s_i\} = 0$. A task system is called a synchronous system if $s_i = s_j$ for all $1 \leq i, j \leq n$; otherwise it is called an asynchronous system. As we shall see later, the complexity of our scheduling problem differs considerably depending upon whether the task system is synchronous or not.

We shall be concerned only with a subclass of preemptive scheduling algorithms which we call fixed-priority algorithms. A distinct (and fixed) priority is assigned to each task according to some mechanism. Then, scheduling is done automatically as follows: At each instant of

---

*This work was supported in part by the National Science Foundation under Grant MCS-79-04898.

time, the processor always executes a request of a higher-priority task, using preemption if necessary, before a request of a lower-priority task. An alternative and more effective way to schedule tasks is to assign priorities on a dynamic basis; i.e. the priority of a task is a function of time. We call this method of scheduling tasks dynamic-priority scheduling. Several authors [3,4,5] have considered dynamic-priority algorithms. However, it is easy to see that dynamic-priority scheduling is more difficult to implement and hence can incur higher system overhead than fixed-priority scheduling. Moreover for the particular computer applications we have in mind, it is possible to implement fixed-priority algorithms in the hardware level by the use of a priority-interrupt mechanism. Not only does this reduce overhead, but makes the time required to switch from one task to another very small. Furthermore, we can adjust the execution time of each task to incorporate this time-loss. This justifies an important assumption, namely that there is no time-loss when the processor is switched from one task to another.

A schedule of a task system is said to be valid if the deadlines of all requests of each task are met. A task system is said to be schedulable (with respect to fixed-priority scheduling) on m processors if there exists a valid schedule for the task system on m processors. A fixed-priority algorithm is said to be optimal for m processors if it produces a valid schedule on m processors for every task system which is schedulable on m processors.

SYNCHRONOUS SYSTEMS

In this section we assume the task system is a synchronous system. For convenience in notation, we assume $s_i = 0$ for all $1 \leq i \leq n$. We shall first consider the single-processor case and then shall consider the multiprocessor case.

Liu and Layland [4] and Serlin [5] have studied fixed-priority scheduling of synchronous systems on one processor under the assumption that $d_i = p_i$ for all $1 \leq i \leq n$. They showed that an optimal fixed-priority algorithm can be obtained by assigning priorities to tasks according to the rate-monotonic priority assignment (Serlin called this the Intelligent Fixed Priority Algorithm). In the rate-monotonic priority assignment, a task is assigned a higher priority if it has a smaller period; i.e. the priority of a task is inversely proportional to its period. If several tasks have the same period, then the tie can be resolved in an arbitrary manner. The rate-monotonic priority assignment however is not optimal if

$d_i \neq p_i$ for some $1 \leq i \leq n$, as the following example shows. Consider the task system consisting of two tasks T_1 and T_2 with $d_1 = p_1 = 2$, $e_1 = 1$, $d_2 = 1$, $p_2 = 3$ and $e_2 = 1$. According to the rate-monotonic priority assignment, T_1 would be assigned a higher priority than T_2. It is easy to verify that the schedule produced by this priority assignment is not valid. However, a valid schedule can be obtained by assigning a higher priority to T_2 than T_1.

In an attempt to devise an optimal priority assignment, we consider the inverse-deadline priority assignment which assigns a higher priority to a task with a smaller deadline (i.e. the priority of a task is inversely proportional to its deadline). If several tasks have the same deadline, then the tie can be broken by an arbitrary tie-breaking rule. It is interesting to observe that the inverse-deadline priority assignment reduces to the rate-monotonic priority assignment if $d_i = p_i$ for all $1 \leq i \leq n$. With regard to the assignment we obtain:

Theorem 1 The inverse-deadline priority assignment is an optimal priority assignment for one processor.

Following characterizations given in [4] we have:

Theorem 2 There is a pseudo-polynomial time algorithm for deciding if the schedule produced by a particular priority assignment is valid.

Theorem 1 shows that we can decide if a synchronous system is schedulable on one processor simply by determining if the schedule produced by the inverse-deadline priority assignment is valid. Since the inverse-deadline priority assignment can be obtained in $O(n \log_2 n)$ time and since we can decide if the schedule produced by the inverse-deadline priority assignment is valid in pseudo-polynomial time (see [2]), we can decide if a synchronous system is schedulable on one processor in pseudo-polynomial time. Thus, we have the following theorem.

Theorem 3 There is a pseudo-polynomial time algorithm for deciding if a synchronous system is schedulable on one processor.

We now turn our attention to multi-processor computing systems. Basically, there are two methods we can schedule tasks on a multi-processor computing system. The first method is that of Dhall and Liu's approach (see [1]) in which tasks are partitioned into separate groups so that each group is assigned to a distinct processor. Tasks within each group are assigned priorities relative to other tasks in the same group. We call this method the partitioning method. The second method, called the nonpartitioning method, is exactly the opposite of the partitioning method in that the tasks are not partitioned into groups. Instead, the m processors are collectively treated as one entity. Each task is assigned a distinct priority and the processors always execute requests of higher-priority tasks before requests of lower-priority tasks.

The partitioning method and the nonpartitioning method are incomparable in effectiveness in the sense that there are task systems for which one method requires less processors than the other, and conversely. This can be seen from the following examples. First, consider the task system consisting of four tasks T_1, T_2, T_3 and T_4 with $d_1 = p_1 = 2$, $e_1 = 1$, $d_2 = p_2 = 4$, $e_2 = 2$, $d_3 = p_3 = 3$, $e_3 = 2$, $d_4 = p_4 = 6$, and $e_4 = 2$. This task system can be scheduled by the partitioning method on two processors by assigning T_1 and T_2 to one processor and T_3 and T_4 to the other. On the other hand, no matter what priority assignment we use, we find that these four tasks cannot be scheduled on two processors by the nonpartitioning method. Next, we consider the task system consisting of four tasks T_1, T_2, T_3 and T_4 with $d_1 = p_1 = 2$, $e_1 = 1$, $d_2 = p_2 = 3$, $e_2 = 1.1$, $d_3 = p_3 = 3$, $e_3 = 2.1$, $d_4 = p_4 = 4$, and $e_4 = 0.8$. If we assign priorities in the order T_1, T_3, T_2, T_4, we find that these four tasks can be scheduled on two processors by the nonpartitioning method. However, they cannot be partitioned into two or less groups so that each group can be scheduled on a distinct processor.

For multi-processors we have the following results:

Theorem 4 Given a synchronous system R and a positive integer m, the problem of deciding whether R is schedulable on m processors with respect to the partitioning method is NP-hard in the strong sense.

Theorem 5 Given a synchronous system R and a positive integer m, the problem of deciding whether R is schedulable on m processors with respect to the nonpartitioning method is NP-hard in the strong sense.

ASYNCHRONOUS SYSTEMS

In this section we study the scheduling of asynchronous systems on $m \geq 1$ processors. Without loss of generality, we assume $\min \{s_i\} = 0$. We shall first consider the single-processor case and then shall consider the multi-processor case. For one processor, the inverse-deadline priority assignment is no longer an optimal priority assignment for asynchronous systems. This can be seen from the task system consisting of two tasks T_1 and T_2 with $s_1 = 1$, $e_1 = 1$, $d_1 = 1.5$, $p_1 = 2$, $s_2 = 0$, $e_2 = 1.5$, $d_2 = 2$ and $p_2 = 4$. It is easy to see that this task system can be scheduled on one processor by the priority assignment $\rho = (T_2, T_1)$. However, the schedule produced by the inverse-deadline priority assignment is not valid.

The example given above only shows that the inverse-deadline priority assignment is not optimal if $d_i \neq p_i$ for some $1 \leq i \leq n$. It leaves open the possibility that it may be optimal if $d_i = p_i$ for all $1 \leq i \leq n$. To foreclose this possibility, we consider the task system consisting of three tasks T_1, T_2 and T_3 with $s_1 = 10$, $e_1 = 1$, $d_1 = p_1 = 12$, $s_2 = 0$, $e_2 = 6$, $d_2 = p_2 = 12$, $s_3 = 0$, $e_3 = 3$ and $d_3 = p_3 = 8$. This task system can be scheduled on one processor by the priority assignment $\rho = (T_3,T_2,T_1)$, and yet the schedule produced by the inverse-deadline priority assignment $\rho' = (T_3,T_1,T_2)$ is not valid.*

At the present time, no priority assignment has been found which is optimal for an arbitrary asynchronous system. However, we can show that the inverse-deadline priority assignment is optimal for two special cases.

Theorem 6 The inverse-deadline priority assignment is an optimal priority assignment for asynchronous systems having only two tasks and $d_i = p_i$ for $i = 1$ and 2.

Theorem 7 The inverse-deadline priority assignment is an optimal priority assignment for asynchronous systems satisfying the conditions that $d_i = p_i$ for all $1 \leq i \leq n$ and p_i is an integral multiple of p_j whenever $p_i > p_j$.

We now turn to the question of deciding whether the schedule produced by a particular priority assignment is valid. Before we present our next result, we need to introduce the following notation. Let S be a schedule of the task system R. We define the configuration of the schedule S for the task system R at time t, denoted by $C_S(R,t)$, to be the n-tuple $(e_{1,t},e_{2,t},\ldots,e_{n,t})$, where $e_{i,t}$ is the amount of time for which task T_i has executed since its last request up until time t, and $e_{i,t}$ is undefined if $t \leq s_i$.

Theorem 8 Let S be the schedule of a task system R constructed by a given priority assignment. S is a valid schedule if and only if (1) all deadlines in the interval $[0,t_2]$ are met in S, where $t_2 = s + 2P$, and (2) $C_S(R,t_1) = C_S(R,t_2)$, where $t_1 = s + P$, $S = \max \{s_i\}$ and P = least common multiple of $\{p_1,\ldots,p_n\}$.

By Theorem 8, an algorithm to determine if the schedule S produced by a particular priority assignment is valid consists of constructing S until time t_2. If all deadlines in $[0,t_2]$ are met in S and $C_S(R,t_1) = C_S(R,t_2)$, then S is a valid schedule; otherwise, it is not. Our algorithm is clearly an exponential time algorithm. Thus, one may wonder whether there is any polynimial time algorithm to solve this problem. Our next result shows that there will not be such an algorithm unless P = NP.

---

*Actually, both ρ and ρ' are inverse-deadline priority assignments since they differ only in the way the tie is resolved for T_1 and T_2. However, we say that the inverse-deadline priority assignment is not optimal if one of them does not produce a valid schedule.

Theorem 9 Given an asynchronous system R and a priority assignment ρ, the problem of deciding if the schedule produced by ρ is valid is NP-hard.

We now turn our attention to the problem of deciding whether a task system is schedulable on one processor. The only method we know of to solve this problem consists of enumerating all n! possible priority assignments and checking to see if the schedule produced by any of these priority assignments is valid. The worst-case running time of this algorithm is $O(n! * (s + P))$, where s and P are defined as above. On the other hand, we probably cannot expect to have a polynomial time algorithm to solve this problem, for we can show this problem is also NP-hard.

Theorem 10 Given an asynchronous system R, the problem of deciding whether R is schedulable on one processor is NP-hard.

Finally we consider the problem of scheduling asynchronous systems on multi-processor computing systems. As before, we can schedule tasks by either the partitioning method or the nonpartitioning method. It is not difficult to show that these two methods are incomparable in effectiveness.

Theorem 11 Given an asynchronous system R and a positive integer m, the problem of deciding whether R is schedulable on m processors with respect to the partitioning method is NP-hard in the strong sense.

Theorem 12 Given an asynchronous system R and a positive integer m, the problem of deciding whether R is schedulable on m processors with respect to the nonpartitioning method is NP-hard in the strong sense.

DISCUSSION

Although we have answered many questions concerning the complexity of scheduling periodic, real-time tasks on $m \geq 1$ processors, a number of important questions remain open. For asynchronous systems, it is clearly desirable to determine the complexity of obtaining an optimal priority assignment for one processor. Also, one question we have not fully addressed, is the problem of deciding whether a particular priority assignment will produce a valid schedule for $m > 1$ processors.

For synchronous systems, we have given a pseudo-polynomial time algorithm to decide if the schedule produced by a particular priority assignment on $m \geq 1$ processors is valid. We have not been able to come up with a polynomial time algorithm to solve this problem, nor can we show this problem is NP-hard. It will be interesting to determine whether this problem is NP-hard or polynomially solvable.

REFERENCES

1. S. K. Dhall and C. L. Liu, "On a real-time scheduling problem," Operations Research, Vol. 26, No. 1, 1978, pp. 127-140.

2. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

3. J. Labetoulle, "Some theorems on real time scheduling," in Computer Architecture and Networks, E. Gelenbe and R. Mahl (eds.), North-Holland Publishing Co., 1974, pp. 285-293.

4. C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," J. ACM, Vol. 20, No. 1, 1973, pp. 46-61.

5. O. Serlin, "Scheduling of time critical processes," Proc. of the Spring Joint Computer Conference, 1972, pp. 925-932.

ON SCHEDULING INDEPENDENT TASKS WITH RESTRICTED EXECUTION TIMES*

JOSEPH Y-T. LEUNG
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

Abstract: We consider the problem of nonpreemptively scheduling n independent tasks on m identical and parallel machines with the objective of minimizing the overall finishing time. The problem has been shown to be NP-complete in the strong sense and hence there probably will not be any pseudo-polynomial time algorithm for solving this problem. We show, however, that if the execution times are restricted to a fixed number, say k, of different values, then it can be solved in polynomial time. Our algorithm runs in time $O(m*\log_2 p*n^{2(k-1)})$ in the worst case, where p is the largest execution time.

INTRODUCTION

We consider one of the most basic and well-studied problems of deterministic scheduling theory, that of nonpreemptively scheduling n independent tasks on m identical and parallel machines (processors) with the objective of minimizing the overall finishing time. Formally, we are given a set $\mathcal{T} = \{T_1,\ldots,T_n\}$ of n tasks, a function $p(\cdot)$ mapping $\mathcal{T}$ into the set of positive integers, and a set of $m \geq 2$ identical and parallel machines. The function $p(\cdot)$ gives the execution time of each task T_i. (We assume the execution times of the tasks are given as integers. This should not cause any great loss of generality since it is equivalent to allowing the execution times to be mutually commensurable.) Tasks are processed nonpreemptively on the machines, i.e., a task once started execution will not be removed from the machine until it has finished execution. The problem is to schedule $\mathcal{T}$ on these m mahcines so that the total time taken to process $\mathcal{T}$ is minimized. Such a schedule is called an optimum schedule.

The problem of finding an optimum schedule has been shown to be NP-complete in the strong sense [5] and hence it is unlikely that a pseudo-polynomial time algorithm (i.e., an algorithm that runs in time bounded by a polynomial function of m, n and $p = \max\{p(T_i)\}$) can be devised to solve the general problem. (The unfamiliar reader is referred to [4,5] for a discussion of the implications of strong NP-completeness.) In this paper we show that the problem can be solved by a polynomial time algorithm (i.e., an algorithm that runs in time bounded by a polynomial function of m, n and $\log_2 p$) if the execution times are drawn from an arbitrary set with fixed cardinality. Our algorithm runs in time $O(m*\log_2 p*n^{2(k-1)})$ in the worst case, where k is the number of different execution times. In practice it is often the case that, while we have a large number of tasks to be scheduled, most of the tasks have the same exectution time. In this case our algorithm finds an optimum schedule relatively quickly.

Recently, there has been some interest in identifying special cases of this problem for which optimim schedules can be found in polynomial time (or even pseudo-polynomial time). If m is fixed, then the algorithm of Sahni [7] can be seen as a pseudo-polynomial time algorithm (although the running time of the algorithm has m as an exponent). Coffman et al. [2] have shown that optimum schedules can be obtained in $O(n \log_2 n)$

---

*This research was supported in part by the National Science Foundation under Grant MCS-79-04898.

time if the execution times can be expressed as powers of some positive real number.

A number of authors have proposed and analyzed fast heuristic algorithms for the general problem. The LPT rule of Graham [6] never produces schedules longer than 4/3 the optimum schedules. Recently, Coffman et al. [2] have shown that optimum schedules can be obtained in $O(n \log_2 n)$ time if the execution times can be expressed as powers of some positive real number.

A number of authors have proposed and analyzed fast heuristic algorithms for the general problem. The LPT rule of Graham [6] never produces schedules longer than 4/3 the optimum schedules. Recently, Coffman et al. [1] proposed and analyzed a modified bin-packing heuristic to solve this problem. They showed the heuristic generates schedules at most 22% longer than optimum schedules. Graham [6] and Sahni [7] have proposed heuristics that can generate schedules arbitrarily close to optimum schedules. However, the running time of the algorithm grows larger as the desired accuracy gets closer to optimum solutions. A survey of these heuristic algorithms can be found in [3].

TWO EXECUTION TIMES

We shall motivate our algorithm by first considering the special case when there are only two execution times. The generalization to k different execution times will be given in the next section. Suppose we are given n tasks with X_1 tasks having execution time A_1 and $X_2 = n - X_1$ tasks having execution time A_2. We wish to schedule these n tasks on m processors so that the overall finishing time is minimized. Let us consider the following decision problem: Given X_1 tasks with execution time A_1 and X_2 tasks with execution time A_2 and positive integers m and D, is there a schedule of these tasks on m processors so that the overall finishing time is D or less? We claim that if there is a polynomial time algorithm to solve the above decision problem, then we can obtain an optimum schedule in polynomial time. This is because we can use the binary search technique to search for the optimum value of D. An obvious lower bound for the optimum value is $L = (X_1A_1 + X_2A_2)/m$. From the analysis of the LPT rule [6], we see that $U = L + p$ is an upper bound for the optimum value, where $p = \max\{A_1,A_2\}$. Thus, in $\log_2 p$ iterations we will be able to determine the optimum finishing time and hence the overall algorithm will run in polynomial time if the above decision problem can be solved in polynomial time.

Our problem is now reduced to solving the above decision problem in polynomial time. This decision problem can be seen as equivalent to the following problem: Given positive integers X_1, A_1, X_2, A_2, m and D, are there integer-valued vectors $Y_1 = (Y_{1,1},\ldots,Y_{m,1})$ and $Y_2 = Y_{1,2},\ldots,Y_{m,2})$ such that (I) $\sum_{i=1}^{m} Y_{i,j} \geq X_j$ for $j = 1$ and 2, and (II) $Y_{i,1}A_1 + Y_{i,2}A_2 \leq D$ for each $1 \leq i \leq m$? For each $1 \leq i \leq m$ and $1 \leq j \leq 2$, $Y_{i,j}$ gives the number of tasks with execution time A_j that are scheduled on the ith processor. Condition (II) ensures that the finishing time of each processor is D or less and condition (i) guarantees that the total number of tasks scheduled is no less than the given number for each category.

To solve the above problem, let us consider the following equation.

$$A_1x + A_2y = D \tag{1}$$

As shown in Figure 1, Equation (1) defines the straight line L in the Euclidean plane. L cuts the x-axis at D/A_1 and the y-axis at D/A_2. Every integer-valued point, say (x',y'), on or below L is a candidate for $Y_{i,1}$ and $Y_{i,2}$, since (x',y') satisfies the condition $A_1x' + A_2y' \leq D$. Thus, all we need do is to pick m of these points so that the sum of the x-coordinates of these m points is at least X_1 and the sum of the y-coordinates is at least X_2. In choosing these points, we only need to consider those points that are at the boundary of L, since every other point is subsumed by one of them. These boundary points are circled in Figure 1. Furthermore, we need not consider those points (x',y') such that $x' > X_1$ or $y' > X_2$. (This can occur if $\lfloor D/A_1\rfloor > X_1$ or $\lfloor D/A_2\rfloor > X_2$.) Now, if we choose these m points by the straight forward enumerative approach, then the running time of the algorithm will be proportional to n^m and hence is exponential to the size of the input, since there may be $O(n)$ boundary points. In the following we will show there is a dynamic programming approach that runs in time $O(mn^2)$ to choose these m points.

Let $B_1 = \min\{X_1,\lfloor D/A_1\rfloor\}$ and $B_2 = \min\{X_2,\lfloor D/A_2\rfloor\}$. Without loss of generality, we may assume $B_1X_1 \le B_2X_2$. For each $0 \le j \le B_1$, we let f_j be the point in the Euclidean plane defined by $f_j = (j,\lfloor (D - jA_1)/A_2\rfloor)$. The set of boundary points that we need to consider is given by $F = \{f_j \mid 0 \le j \le B_1\}$. The heart of our algorithm is to construct a 2-dimensional table S having m rows and $X_1 + 1$ columns. The rows are numbered from 1 to m and the columns are numbered from 0 to X_1. For each $1 \le i \le n$ and $0 \le j \le X_1$, $S(i,j)$ gives the largest y-coordinate that can be obtained by choosing i boundary points from F such that the sum of the x-coordinates of these i points is j and the sum of the y-coordinates of these i points is $S(i,j)$. For convenience, we let $S(i,j)$ to be -1 if we cannot find i boundary points from F such that the sum of the x-coordinates is j. It is now easy to see that there are integer-valued vectors $Y_1 = (Y_{1,1},\ldots,Y_{m,1})$ and $Y_2 = (Y_{1,2},\ldots,Y_{m,2})$ satisfying conditions (I) and (II) if and only if $S(m,X_1) \ge X_2$.

To construct the table S, we proceed row by row. The first row can easily be determined since $S(1,j) = \lfloor (D - jA_1)/A_2\rfloor$ if $j \le B_1$ and $S(1,j) = -1$ if $j > B_1$. That is, the first $B_1 + 1$ entries of the first row are given by the y-coordinates of the boundary points $f_j (0 \le j \le B_1)$. Suppose the first i rows of S have been computed. We can compute the $(i + 1)$st row of S as follows: For each $0 \le j \le X_1$, we let $N_{i+1,j} = \{S(i,j - k) + S(1,k) \mid 0 \le k \le B_1,\ j - k \ge 0,\ S(i,j - k) \ne -1\}$. $S(i + 1,j)$ is then given by the largest number in $N_{i+1,j}$ if $N_{i+1,j} \ne \emptyset$ and $S(i + 1,j) = -1$ if $N_{i+1,j} = \emptyset$. To illustrate this computation, we consider the following example.

Example 1. Let $X_1 = 11$, $A_1 = 7$, $X_2 = 9$, $A_2 = 5$, $m = 6$ and $D = 22$. $B_1 = \min\{11,\lfloor 22/7\rfloor\} = 3$ and $B_2 = \min\{9,\lfloor 22/5\rfloor\} = 4$. In this case, X_1B_1 is less than X_2B_2. The table S is shown in Table I. The first four entries of the first row are set to the y-coordinates of the boundary points $f_j (0 \le j \le 3)$ respectively. The remaining entries of the first row are set to -1. Let us consider the computation of the second row. To compute $S(2,2)$, for example, we first determine the set $N_{2,2}$. $N_{2,2} = \{S(1,2) + S(1,0),\ S(1,1) + S(1,1),\ S(1,0) + S(1,2)\} = \{5,6,5\}$. Thus, $S(2,2) = 6$. $N_{2,3} = \{S(1,3) + S(1,0),\ S(1,2) + S(1,1),\ S(1,1) + S(1,2),\ S(1,0) + S(1,3)\} = \{4,4,4,4\}$ and hence $S(2,3) = 4$. $N_{2,5} = \{S(1,3) + S(1,2),\ S(1,2) + S(1,3)\} = \{1,1\}$. Notice that $S(1,5) + S(1,0)$ and $S(1,4) + S(1,1)$ are not in $N_{2,5}$ since both $S(1,5)$ and $S(1,4)$ are equal to -1. In this case, $S(2,5) = 1$. Finally, $N_{2,7} = \emptyset$ since $S(1,7)$, $S(1,6)$, $S(1,5)$ and $S(1,4)$ are all equal to -1. Thus, $S(2,7) = -1$. From Table I, we see that $S(6,11) > 9$. Thus, there are integer-valued vectors $Y_1 = (Y_{1,1},\ldots,Y_{6,1})$ and $Y_2 = (Y_{1,2},\ldots, Y_{6,2})$ such that $\sum_{i=1}^{6} Y_{i,1} \ge 11$, $\sum_{i=1}^{6} Y_{i,2} \ge 9$ and $7Y_{i,1} + 5Y_{i,2} \le 22$ for each $1 \le i \le 6$.

The table S only gives us the answer whether or not there are integer-valued vectors Y_1 and Y_2 satisfying conditions (I) and (II). It does not, however, give us the actual values of the two vectors. In order to obtain these values, we can store a pointer in every entry of the table which points to the previous row from where we obtain the present entry. Using these pointers, we can determine the actual values of the two vectors Y_1 and Y_2. In Table I, we show the sequence of pointers used to obtain the final entry S(6,11). Tracing these pointers from S(6,11) , we obtain the vectors $Y_1 = (0,1,1,3,3,3)$ and $Y_2 = (4,3,3,0,0,0)$.

We now analyze the time and space requirements of the algorithm. The table S has $m(X_1 + 1)$ entries. To compute each entry, we spend time at most $B_1 + 1$ steps. Thus, the total time needed to compute the whole table is at most $m(X_1 + 1)(B_1 + 1)$ steps. Since X_1 and B_1 are both bounded above by n, the worst-case running time of the algorithm is $O(mn^2)$. The space requirement of the algorithm is $O(mn)$.

THE GENERAL ALGORITHM

The ideas in the last section can be generalized to $k > 2$ different execution times. In this case, we are given positive integers $X_1,A_1,\ldots,X_k,A_k,m$ and D, and we want to determine whether there are integer-valued vectors $Y_1 = (Y_{1,1},\ldots,Y_{m,1}),\ldots,Y_k = (Y_{1,k},\ldots,Y_{m,k})$ such that $\sum_{i=1}^{m} Y_{i,j} \geq X_j$ for each $1 \leq j \leq k$ and $\sum_{j=1} Y_{i,j}A_j \leq D$ for each $1 \leq i \leq m$. The equation $A_1x_1 + A_2x_2 + \ldots + A_kx_k = D$ defines a surface in the k-dimensional space. We need to pick m boundary points on or below the surface so that the sum of the jth-coordinates of these m points is at least X_j for each $1 \leq j \leq k$. Let $B_j - \min\{X_m k\lfloor D/A_j\rfloor\}$ for each $1 \leq j \leq k$. We assume $B_kX_k = \max_{1\leq j\leq k}\{B_jX_j\}$. (If this is not true, we can reindex them so that $B_kX_k = \max_{1\leq j\leq k}\{B_jX_j\}$.) We will construct a k-dimensional table S with size m in the first dimension and size $X_j + 1$ in the $(j + 1)$st dimension for each $1 \leq j \leq k - 1$. For each $1 \leq i \leq m$, $0 \leq j_1 \leq X_1,\ldots,0 \leq j_{k-1} \leq X_{k-1}$, $S(i,j_1,\ldots,j_{k-1})$ gives the largest kth-coordinate that can be obtained by choosing i boundary points such that the sum of the ℓth-coordinates of these i points is j_ℓ for each $1 \leq \ell \leq k - 1$ and the sum of the kth-coordinates of these i points is $S(i,j_1,\ldots,j_{k-1})$. We define $S(i,j_1,\ldots,j_{k-1})$ to be -1 if we cannot find i boundary points such that the sum of the ℓth-coordinates of these i points is j_ℓ for each $1 \leq \ell \leq k -1$. The whole algorithm of finding the optimum finishing time is given as follows.

```
procedure OPTFIN
comment   This algorithm computes the optimum finishing time of a set of
          tasks with k different execution times to be scheduled on m pro-
          cessors. The arrays A and X store the execution time and the
          number of tasks of each category respectively. The algorithm
          employs a function DEADLINE which takes the parameter D and
          determines whether there is a schedule with finishing time D or
          less. The algorithm performs a binary search over a range of
          possible values to determine the optimum finishing time. All
          variables used are integer variables; integer divisions are
          assumed throughout the algorithm.
function  DEADLINE(D)
    comment   DEADLINE is a logical function. It returns the value "true"
              if there is a schedule with finishing time D or less, and
              "false" otherwise. The array B is used to store
              min{X_j,⌊D/A_j⌋} for each 1 ≤ j ≤ k. The table S is
              k-dimensional table.
```

```
        for j ← 1 to k do
            B[j] ← min{X[j],D/A[j]};
        reindex X[j]and B[j] so that $B[k]X[k] = \max_{1\le j\le k} \{B[j]X[j]\}$;
        for $j_1$ ← 0 to X[1] do
            for $j_2$ ← 0 to X[2] do
                .
                .
                .
                for $j_{k-1}$ ← 0 to X[k-1] do
                    begin
                        T ← D - $\sum_{\ell=1}^{k-1} j_\ell$*A[ℓ];
                        if T < 0 then S[1,$j_1$,...,$j_{k-1}$] ← -1
                                 else S[1,$j_1$,...,$j_{k-1}$] ← T/A[k]
                    end;
        if S[1,X[1],...,X[k - 1]] ≥ X[k] then return ("true");
        for i ← 2 to m do
            begin
                for $j_1$ ← 0 to X[1] do
                    for $j_2$ ← 0 to X[2] do
                        .
                        .
                        .
                        for $j_{k-1}$ ← 0 to X[k - 1] do
                            begin
                                for ℓ ← 1 to k - 1 do
                                    $n_\ell$ ← max{0,$j_\ell$ -B[ℓ]};
                                S[i,$j_1$,..., $j_{k-1}$] ← -1;
                                for $\ell_1$ ← $n_1$ to $j_1$ do
                                    for $\ell_2$ ← $n_2$ to $j_2$ do
                                        .
                                        .
                                        .
                                        for $\ell_{k-1}$ ← $n_{k-1}$ to $j_{k-1}$ do
                                            if S[i-1,$\ell_1$,...,$\ell_{k-1}$] ≥ 0 then
                                            S[i,$j_1$,...,$j_{k-1}$]
                                              ← max {S[i,$j_1$,...,$j_{k-1}$],
                                            S[i-1,$\ell_1$,...,$\ell_{k-1}$]
                                              + S[1,$j_1$-$\ell_1$,...,$j_{k-1}$-$\ell_{k-1}$]}
                            end;
                if S[i,X[1],...,X[k - 1]] ≥ X[k] then return ("true")
            end
        if S[m,X[1],...,X[k - 1]] ≥ X[k] then return ("true")
                                         else return ("false")
    end DEADLINE
    read (m);
    for j ← 1 to k do read (X[j], A[j]);
    LOW ←($\sum_{j=1}^{k}$ X[j]*A[j])/m;
```

```
HIGH ← LOW + $\max_{1 \le j \le k} \{A[j]\}$;
while LOW < HIGH do
    begin
        MID ← (LOW + HIGH)/2;
        if DEADLINE(MID) then HIGH ← MID
                         else LOW ← MID + 1
    end;
print ("The optimum finishing time is", HIGH)
end OPTFIN
```

The algorithm we gave only determines the optimum finishing time. If we are interested in finding the actual schedule, we need to modify the function DEADLINE so that it stores an additional pointer in the table S. In the main procedure, after the optimum finishing time had been determined, we call DEADLINE once more with D = HIGH. This will regenerate S for the optimum finishing time. Following the pointers stored in S, we can determine the vectors $Y_1,\ldots,Y_k$.

We now analyze the time and space requirements of OPTFIN. The table S has $m*\prod_{j=1}^{k-1}(X_j+1)$ entries. Each entry can be computed in $O(\prod_{j=1}^{k-1}(B_j+1))$ steps. Thus, the total time taken for each call of DEADLINE is $O(m*\prod_{j=1}^{k-1}[(X_j+1)(B_j+1)])$. Since DEADLINE is called $\log_2 p$ times and since X_j and B_j are bounded above by n, the worst-case running time of OPTFIN is $O(m*\log_2 p*n^{2(k-1)})$, where p is the largest execution time. The space used by OPTFIN is readily seen to be $O(m*n^{k-1})$ in the worst case.

DISCUSSIONS

The running time of our algorithm can be improved if tighter lower and upper bounds for the optimum finishing time can be obtained. In this regard, we can incorporate a preprocessing step into our algorithm by running some of the heuristic algorithms discussed in [3] and choosing the smallest finishing time found for the upper bound. The running time of our algorithm can also be reduced if there is a quicker way to compute each entry of the table S. It is conceivable that some characterizations exist which allow us to compute each entry by examining fewer possibilities. Further research along this line is needed.

If we are only interested in finding the optimum finishing time, the space needed by our algorithm can be reduced to $O(n^{k-1})$ without substantial increase in running time. Let $S'_i = \{S(i,j_1,\ldots,j_{k-1}) \mid 0 \le j_1 \le X_1,\ldots,0 \le j_{k-1} \le X_{k-1}\}$ for each $1 \le i \le m$. Observe that the algorithm only needs the set S'_m. Since S'_i can be generated from S'_{i-1} for each $2 \le i \le m$, we can successively generate S'_i using only two (k - 1)-dimensional tables, say P and Q. P will be used to store S'_i for i odd and Q will be used to store S'_i for i even. At the end of the algorithm, only the sets S'_{m-1} and S'_m will appear in P and Q. This approach does not work well if we are interested in finding the actual schedule since we cannot save the pointers. However, if space is a limiting factor, we can still use this approach at the expense of an increase in running time. This can be done as follows. After S'_{m-1} and S'_m have been generated, we can determine the entry in S'_{m-1} from where we obtain the entry $S(m,X_1,\ldots,X_{k-1})$.

We save all information concerning this entry. We then regenerate the sets S'_{m-3} and S'_{m-2} using the tables P and Q. Using the previously saved information, we can backtrack to S'_{m-3}. This process is iterated until we reach S'_1. The time taken by this process is $O(m^2 * n^{2(k-1)})$ in the worst case. Thus, the worst-case running time of the algorithm becomes $O((\log_2 p + m) * m * n^{2(k-1)})$.

Finally, for future research directions, it is highly desirable to investigate the practicality of this algorithm in real-life situations.

REFERENCES

[1] E. G. Coffman, Jr., M. R. Garey and D. S. Johnson, "An application of bin-packing to multiprocessor scheduling," SIAM J. on Computing, Vol. 7, No. 1, 1978, pp. 1-17.

[2] E. G. Coffman, Jr., J. Y-T. Leung and D. Slutz, "On the optimality of fast heuristics for scheduling and storage allocation problems," Foundations of Control Engineering, Vol. 3, No. 4, 1978, pp. 161-169.

[3] M. R. Garey, R. L. Graham and D. S. Johnson, "Performance guarantees for scheduling problems," Operations Research, Vol. 26, No. 1, 1978, pp. 3-21.

[4] M. R. Garey and D. S. Johnson, "Strong NP-completeness results: motivation, examples, and implications," J. of ACM, Vol. 25, No. 3, 1978, pp. 499-508.

[5] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[6] R. L. Graham, "Bounds on multiprocessing timing anomalies," SIAM J. on Applied Math, Vol. 17, No. 2, 1969, pp. 416-429.

[7] S. Sahni, "Algorithms for scheduling independent tasks," J. of ACM, Vol. 23, No. 1, 1976, pp. 116-127.

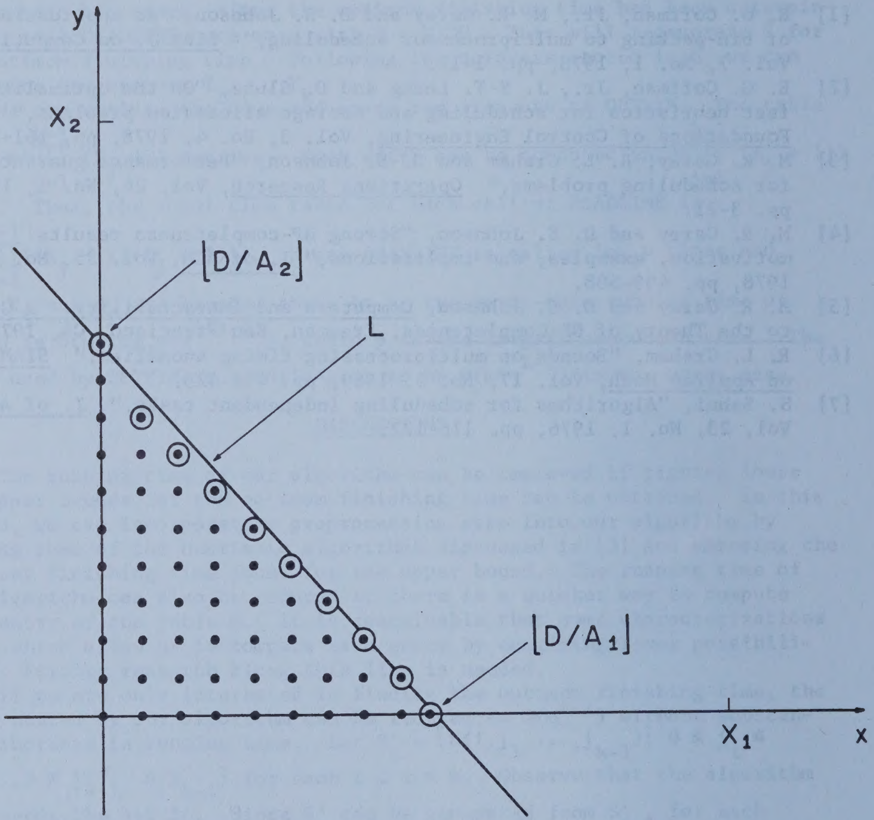
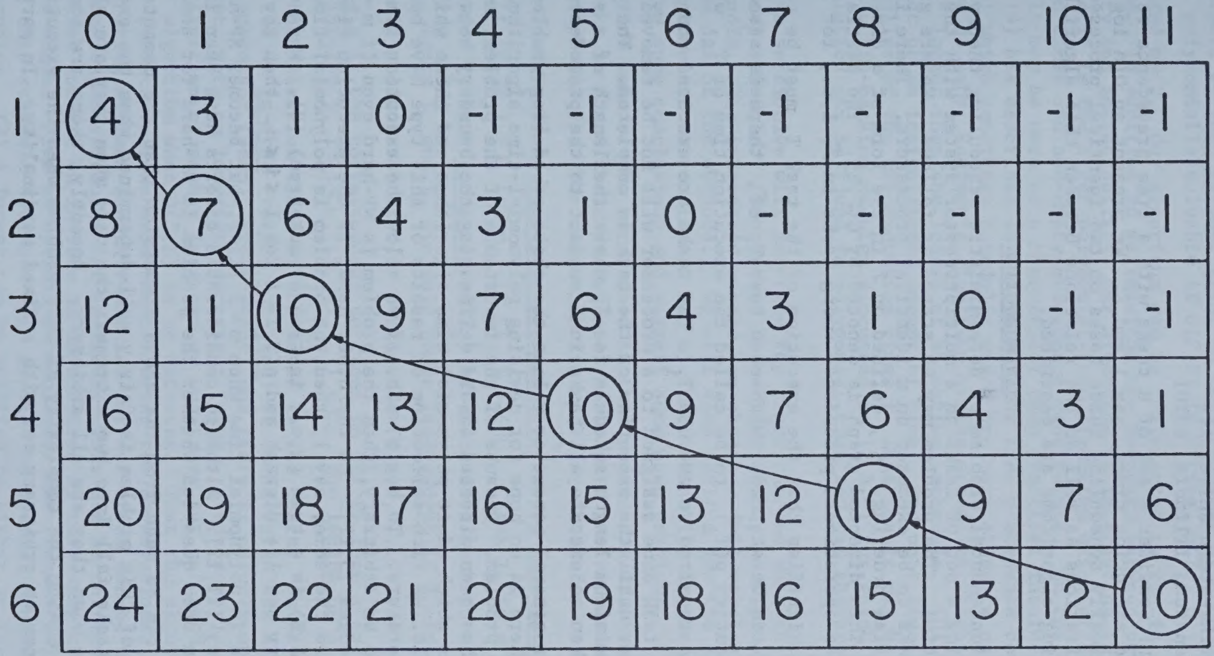

y
X_2
$[D/A_2]$
L
$[D/A_1]$
X_1
x

Figure 1. The straight line given by the equation $A_1 x + A_2 y = D$.

Table I. The table S in Example 1. $X_1 = 11$, $A_1 = 7$, $X_2 = 9$, $A_2 = 5$, $m = 6$ and $D = 22$.



	0	1	2	3	4	5	6	7	8	9	10	11
1	4	3	1	0	-1	-1	-1	-1	-1	-1	-1	-1
2	8	7	6	4	3	1	0	-1	-1	-1	-1	-1
3	12	11	10	9	7	6	4	3	1	0	-1	-1
4	16	15	14	13	12	10	9	7	6	4	3	1
5	20	19	18	17	16	15	13	12	10	9	7	6
6	24	23	22	21	20	19	18	16	15	13	12	10

OPTIMAL TWO PROCESSOR SCHEDULING OF TREE PRECEDENCE CONSTRAINED TASKS WITH TWO EXECUTION TIMES*

K. NAKAJIMA, J.Y-T. LEUNG AND S. L. HAKIMI
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

Abstract: Consider a set of n tasks with a tree precedence relation and two possible execution times, say 1 and 2. We present an O(n log n) algorithm for optimally scheduling these tasks on two identical processors. Possible generalizations as well as the relations between this algorithm and other scheduling algorithms are described.

INTRODUCTION

A fundamental problem of deterministic scheduling theory is that of scheduling a set of tasks on a multiprocessor system with the minimum finishing time. The problem may be stated as follows. We are given a set $\mathcal{J}$ of n tasks to be executed on m identical processors. There is an irreflexive partial ordering, "$<$", defined on $\mathcal{J}$ in the form of a directed acyclic graph. This directed graph is denoted by $G(\mathcal{J},E)$. The partial order, $<$, specifies operational precedence among tasks in $\mathcal{J}$, i.e., for T_i and $T_j \in \mathcal{J}$, $T_i < T_j$ signifies that the execution of the task T_i must be completed before T_j can be started. With each task $T_i \in \mathcal{J}$, there is associated a positive quantity $p(T_i)$, to be called the execution time of T_i, which represents the time required to execute T_i. The tasks are executed nonpreemptively; i.e., a task once assigned to a processor will not be removed from that processor until the execution of the task is completed. The problem is to find a minimum length schedule for $\mathcal{J}$, where the length of a schedule is the time taken to execute all tasks in $\mathcal{J}$ subject to the precedence constraints.

For general parameter values the above scheduling problem is NP-hard. Thus, there is no hope for devising polynomial-time algorithms to solve the general problem. Because of the importance of the problem, much research effort has been directed toward delineating the boundary between the subproblems which admit polynomial-time algorithms and those which are intractable. A rich collection of results of this type have been reported in the literature. To begin with, if we allow the execution times of the tasks to be arbitrary, then the problem is NP-hard even if $m = 2$ and $G(\mathcal{J},E)$ has no edges [5,12]. On the other hand, if the execution times of all tasks are the same, say 1, then the problem is polynomial-time solvable when $G(\mathcal{J},E)$ is a tree (i.e., in-tree or out-tree) [8]. When $G(\mathcal{J},E)$ is an arbitrary acyclic digraph and $p(T_i) = 1$ for $1 \leq i \leq n$, then the problem is solvable in polynomial time when $m = 2$ [1,4], but becomes NP-hard if m is arbitrary [11,12]. Although considerable efforts have been invested, it is still an open question whether the problem is NP-hard for some fixed $m > 2$.

It can be seen from the above discussion that the computational complexity of this problem is fairly well-determined when the execution time of the tasks fall into two extremes: the case when they are all equal and the case when they are all arbitrary. Recently, there are some interests in determining the complexity of this problem when the execution times are drawn from an arbitrary set with a fixed cardinality. In particular in

---

*This work was supported by National Science Foundation Grants ENG79-09724 and MCS-79-04898.

such a case, if $G(\mathcal{T},E)$ has no edges, then the problem is solvable in polynomial time [10]. On the other hand, if $G(\mathcal{T},E)$ is an arbitrary acyclic digraph, then the problem is NP-hard even when $m = 2$ and $p(T_i) \in \{1,2\}$ for $i = 1,2,\ldots,n$, [11,12].

In this paper, we shed further light on this problem, by showing that the problem is polynomially solvable if $G(\mathcal{T},E)$ is a tree, $p(T_i) \in \{1,2\}$ for $i = 1,2,\ldots,n$, and $m = 2$. Our algorithm can be implemented to run in $O(n \log n)$ time. In addition to delineating the boundary between polynomial solvability and intractability, our algorithm is interesting in its own right since it is the first polynomial-time algorithm which allows unequal execution times and non-empty precedence constraints. On the other hand, our result may be more of a theoretical than practical interest, since Kaufman [9] has described an algorithm (for tree precedence constrained tasks) which generates schedules which are very close to optimal. For the problem at hand, Kaufman's algorithm yields schedules which are at most one "unit" longer than the length of an optimal schedule. (Of course, one unit may not be an insignificant amount of time. One unit time is here in fact the smaller of the two execution times.) This result may be contrasted with the bin-packing problem discussed in [2,3,6] for which a good approximate solution can be found in polynomial time and yet the problem is NP-hard.

In the next section we will describe the algorithm and show that it yields an optimal schedule. (Because of space limitation, we will omit most of the proofs. The interested reader can obtain a full version of this paper from one of the authors.) We will also outline how the algorithm can be implemented to run in $O(n \log n)$ time. Finally, in the last section, we draw some conclusions and suggest some open problems.

We now define some of the terms used in this paper. As was mentioned earlier the tasks in $\mathcal{T}$ and their precedence constraints are represented by an acyclic digraph $G(\mathcal{T},E)$. The execution time $p(T_i) \in \{1,2\}$ for $i = 1,2,\ldots,n$. Task T_i is an immediate predecessor (resp., immediate successor) of T_j if $(T_i,T_j) \in E$ (resp., if $(T_j,T_i) \in E$). Task T_k is a predecessor of T_ℓ (or T_ℓ is a successor of T_k) if there is a path in $G(\mathcal{T},E)$ from T_k to T_ℓ. Tasks T_k and T_ℓ are said to be independent if T_k is neither a successor nor a predecessor of T_ℓ. A sequence $(T_{i_1},T_{i_2},\ldots,T_{i_k})$ of tasks is called a chain if the corresponding vertices in $G(\mathcal{T},E)$ form a path, i.e., for any integer j, $1 \leq j < k$, $(T_{i_j},T_{i_{j+1}}) \in E$. The length of a chain is the sum of the execution times of the tasks in the chain. An acyclic digraph is called an in-tree (resp., out-tree) if each vertex has at most one immediate successor (resp., predecessor) and there is exactly one vertex, called root, with no immediate successor (resp., predecessor). A forest is a collection of either in-trees or out-trees but not both. We will state our algorithm when $G(\mathcal{T},E)$ is an out-tree. That the algorithm also works for forests follows from the following two observations. (1) An algorithm generating minimum-length schedules for out-trees can be used to generate minimum-length schedules for in-trees. This can be carried out as follows. First convert an in-tree to an out-tree by reversing the arrowheads on the edges. Find an optimal schedule for the (converted) out-tree. Reading this schedule backward will provide an optimal schedule for the original problem. (2) A forest can be modified to either an in-tree or an out-tree by adding a dummy task (vertex) T_0 to $\mathcal{T}$ and joining it to the roots of the forest. An optimal schedule for the modified (in-tree or out-tree) task system gives an optimal schedule for the original forest. Thus, without loss of generality, we will concentrate on out-trees in the remainder of this paper and for convenience we refer to an out-tree as simply a tree. In a tree, the vertex that has no predecessor is called the root and the vertices that have no successors are called the leaves. Given the tree $G(\mathcal{T},E)$, let $S(T_i) = \{T_j \in \mathcal{T} | T_j$ is either T_i or is a successor $T_i\}$. We let $W(T_i)$ denote the sum of the execution times of the tasks in $S(T_i)$. For $T_j \in S(T_i)$, $D(T_i,T_j)$ denotes the sum of the execution times of the tasks in the path from T_i to T_j. In general, the cardinality of a set A is denoted by $|A|$. Figure 1 illustrates the concepts just defined.

THE ALGORITHM

We are given a set of n tasks $\mathcal{T}$ and the precedence relation given by an acyclic digraph $G(\mathcal{T},E)$ which is a tree (out-tree). The algorithm first orders the tasks in the form of a list $L = (T_1,T_2,\ldots,T_n)$ such that for each i, $1 \le i < n$, we have (1) $W(T_i) \ge W(T_{i+1})$, and (2) if $W(T_i) = W(T_{i+1})$, then $|S(T_i)| \ge |S(T_{i+1})|$. Then, this list is used to construct a schedule in the following manner. Whenever a processor becomes free, the list is scanned from left to right and the first unexecuted and "ready" task encountered will be assigned to that processor. (A task is ready if all of its predecessors have been completed.) A schedule thus constructed will be called a Largest-Subtree-First (or LSF) schedule. We will identify the two processors by P_1 and P_2 and we will adopt the convention that when P_1 and P_2 become free simultaneously, P_1 is always the first processor to be assigned a task by the algorithm.

Our algorithm employs the LSF strategy as a subroutine and works as follows. It first constructs an LSF schedule for the whole tree. The schedule so obtained is in fact a minimum-length schedule in almost all cases, (see Fig. 2 for some notable exceptions). It will be shown later that when the LSF schedule is not optimal, then the processors P_1 and P_2 execute three 2-unit leaf tasks in the last four time units of the schedule. We call this a degenerate case. When the degenerate case occurs, the algorithm scans the schedule backward and locates a certain task Y (which finishes execution at time t^*) such that every task that is executed beyond t^* belongs to $S^-(Y) = S(Y) - \{Y\}$. If all tasks in $S^-(Y)$ are 2-unit tasks, then the present schedule is an optimal schedule and the algorithm terminates. Otherwise, it repairs the schedule beyond t^* by first pruning the subtree rooted at Y. This pruning involves iterative removal of pairs of 2-unit leaf tasks farthest from Y until a 1-unit task is exposed (for the first time) as a leaf. Then the algorithm proceeds by using the LSF strategy to construct a schedule for the pruned subtree. Finally the removed tasks are tacked on the back of the schedule for the pruned subtree in the reverse order of their removal. The desired optimal schedule for $\mathcal{T}$ is then the LSF schedule up to t^* followed by the repaired schedule for $S^-(Y)$. We now proceed with a precise statement of our algorithm and the proof of its validity.

Our first goal is to establish that the LSF schedule is an optimal schedule in all but the degenerate case. Let $\sigma(\mathcal{T})$ be an LSF schedule for the tasks in $\mathcal{T}$. Let for each $T_i \in \mathcal{T}$, $s(T_i)$ and $f(T_i)$ denote the starting and finishing times for task T_i in the schedule $\sigma(\mathcal{T})$, respectively. Let f^* be the finishing time of the schedule, i.e., $f^* = \max\{f(T_i) \mid 1 \le i \le n\}$. A time interval $I = (t',t'')$ is an idle interval in $\sigma(\mathcal{T})$ if $0 \le t' < t'' \le f^*$ and

I is a maximal interval during which one of the processors is free. The interval I is called an *internal idle interval* if $t'' < f^*$; otherwise it is called an *external idle interval*. Let $\sigma(\mathcal{T})$ have k internal idle intervals, $I_j = (t'_j, t''_j)$, $j = 1,2,\ldots,k$, and note that $t''_j < t'_{j+1}$, $1 \le j \le k-1$. In what follows t''_0 is defined to be 0. We now divide $\sigma(\mathcal{T})$ into $k+1$ segments, $\sigma_j(\mathcal{T})$, $j = 1,2,\ldots,k+1$, where $\sigma_j(\mathcal{T})$ is the subschedule of $\sigma(\mathcal{T})$ consisting of the tasks in $\{T_i \in \mathcal{T} \mid t''_{j-1} \le s(T_i) < t''_j\}$ for $j = 1,2,\ldots,k$ and $\sigma_{k+1}(\mathcal{T})$ is the subschedule of $\sigma(\mathcal{T})$ consisting of the tasks in $\{T_i \in \mathcal{T} \mid t''_k \le s(T_i)\}$. In an abuse of notation we will also allow $\sigma_j(\mathcal{T})$ to represent the set of tasks in the schedule $\sigma_j(\mathcal{T})$. Figure 3 shows an LSF schedule for the task system shown in Fig. 1 and illustrates some of the concepts just defined.

Lemma 1. *For each* j, *there is a chain of tasks in* $\sigma_j(\mathcal{T})$ *with length* $t''_j - t''_{j-1}$, $1 \le j \le k$.

Proof. Since one of the processors is free in the time interval (t'_j, t''_j) and since the LSF strategy considers all ready tasks for execution whenever a processor is free, there is a unique task $T_{j,1}$ which finishes at t''_j and is a predecessor of all tasks in $\bigcup_{i=j+1}^{k+1} \sigma_i(\mathcal{T})$. $T_{j,1}$ is a task in the desired chain, $1 \le j \le k$. From the definition of $\sigma_i(\mathcal{T})$, $2 \le i \le k+1$, we have $|\sigma_i(\mathcal{T})| \ge 2$ and hence $W(T_{j,1}) \ge p(T_{j,1}) + 2$, $1 \le j \le k$.

Assume, by induction, that there is a chain of tasks $(T_{j,\ell}, T_{j,\ell-1}, \ldots, T_{j,1})$ executed in the time interval $(s(T_{j,\ell}), t''_j)$ such that $\sum_{k=1}^{\ell} p(T_{j,k}) = t''_j - s(T_{j,\ell})$. We wish to show that the immediate predecessor of $T_{j,\ell}$ must finish at time $s(T_{j,\ell})$. Suppose otherwise, and assume that the immediate predecessor of $T_{j,\ell}$ finished at time $\bar{t} < s(T_{j,\ell})$. Since $T_{j,\ell}$ was ready for execution at $\bar{t}$ but was not assigned to a processor until $s(T_{j,\ell})$, both processors must be busy in the interval $(\bar{t}, s(T_{j,\ell}))$. Among the tasks executed in the interval $(\bar{t}, s(T_{j,\ell}))$ select a task T' such that $f(T') = s(T_{j,\ell})$. We note that T' and $T_{j,\ell}$ are independent, then by the LSF strategy, we have $W(T') \ge W(T_{j,\ell})$. However, as there is the idle interval (t'_j, t''_j), all successors of T' must be executed in the interval $(s(T_{j,\ell}), t'_j)$. This implies that $W(T') \le p(T') + t'_j - s(T_{j,\ell})$. On the other hand, $W(T_{j,\ell}) \ge t''_j - s(T_{j,\ell}) + (W(T_{j,1}) - p(T_{j,1})) \ge t''_j - s(T_{j,\ell}) + 2$. The previous two inequalities and the facts that $W(T') \ge W(T_{j,\ell})$ and $t'_j \le t''_j - 1$ imply that $p(T') \ge 3$. This is a contradiction, hence the lemma is proved.

Corollary 1. *There is a chain of tasks in* $\bigcup_{j=1}^{k} \sigma_j(\mathcal{T})$ *with length* t''_k.

Proof. The proof immediately follows from Lemma 1 and the fact that $T_{j,1}$, $1 \le j \le k$, is a predecessor of all tasks in $\sigma_{j+1}(\mathcal{T})$.

Corollary 1 implies that $T_{k,1}$ cannot finish earlier than t''_k in any schedule. Since $T_{k,1}$ does finish at t''_k in an LSF schedule, and since $T_{k,1}$ precedes every task in $\sigma_{k+1}(\mathcal{T})$, it follows that an LSF schedule is an optimal schedule if the tasks in $\sigma_{k+1}(\mathcal{T})$ are scheduled optimally by the LSF strategy. The next lemma shows that an LSF schedule is a minimum-length schedule in all but the "degenerate case". The degenerate case is defined as follows. Let $\sigma(\mathcal{T})$ be an LSF schedule with finishing time f^*. Then the schedule is *degenerate* if P_1 executes two 2-unit tasks from f^*-4 to f^*, and P_2 executes a 2-unit task from f^*-4 to f^*-2 and is idle from f^*-2 to f^*; and furthermore these three 2-unit tasks are all leaf tasks.

Lemma 2. If an LSF schedule, $\sigma(\mathcal{T})$, of the tasks in $\mathcal{T}$ is not degenerate, then $\sigma(\mathcal{T})$ is a minimum-length schedule for $\mathcal{T}$.

If the degenerate case does not occur, our algorithm terminates with LSF schedule $\sigma(\mathcal{T})$ being the final schedule. If the degenerate case does occur, then our LSF schedule may still be a minimum-length schedule. The algorithm carries out one more step before it starts the repairing process. It scans the LSF schedule $\sigma(\mathcal{T})$ backward from f^* to t''_k. Let X, if possible, be the first task encountered in this scan such that $W(X) = 1$ (i.e., X is a 1-unit leaf task). Assume X is executed in the unit time interval (t^*-1, t^*) and let Y be the other task being executed in the interval (t^*-1, t^*). (It will be shown later that $f(Y) = t^*$.) If X does not exist, then let Y be $T_{k,1}$ and $t^* = f(T_{k,1}) = t''_k$. Let R^* and $S^* = S^-(Y)$ denote the set of tasks executed before t^* and after t^*, respectively. It is easy to see that the sum of the execution times of the tasks in S^* is even, and hence S^* must contain an even number of 1-unit tasks none of which are leaf tasks. For example, in the schedule shown in Fig. 3, $t^* = 10$, $Y = T_{11}$, $S^* = \{T_i \mid 12 \le i \le 18\}$, and $X = T_{10}$. As indicated in the following lemma, we are mainly interested in the task Y and the set S^*.

Lemma 3. Let X, t^*, Y, and S^* be defined as above. Then we have: (1) every task in S^* is a successor of Y, and (2) a chain of tasks of length t^* with Y being the last task in the chain.

As a consequence of Lemma 3, we see that a minimum-length schedule for S^* when appended to the original (LSF) schedule for R^* will yield an overall minimum-length schedule for $\mathcal{T}$. Thus, our problem is reduced to finding a minimum-length schedule for S^*. If S^* contains only 2-unit tasks, then the present schedule for the tasks in S^* is already optimal, consequently the LSF schedule $\sigma(\mathcal{T})$ is the desired schedule. Thus we will consider the case when S^* has some (by necessity an even number of) 1-unit tasks.

Let G_0 denote the subtree of $G(\mathcal{T},E)$ rooted at Y. From the previous discussion, we know that all leaf tasks in G_0 are 2-unit tasks, and by the definition of the degenerate case, there are at least three leaves in G_0. Also G_0 must contain at least two (non-leaf) 1-unit tasks. Our algorithm prunes G_0 until a 1-unit task is exposed as a leaf. More precisely, our procedure is described as follows. Let E_1 and F_1 be two leaf tasks farthest away from Y, i.e., $D(Y,E_1)$ and $D(Y,F_1) \ge D(Y,T)$ for every other leaf task T in G_0. Our algorithm first removes E_1 and F_1 from G_0 to obtain G_1. If G_1 contains a 1-unit leaf task, the pruning process is terminated; otherwise it proceeds as above to remove E_2 and F_2 from G_1 to form G_2. This process is repeated until for some r, G_r contains a 1-unit leaf task. (As we will see in the next lemma, in this pruning process, we never encounter a situation where the only leaf task is a single 2-unit leaf task.)

Figure 4 illustrates the iterations involved in the pruning of the subtree rooted at T_{11}. Thus the tasks E_j and F_j denote the tasks removed at the jth iteration of the pruning process, and for convenience, we assume $D(Y,E_j) \geq D(Y,F_j)$, $1 \leq j \leq r$.

The next step in our algorithm involves scheduling the tasks in G_r using the LSF strategy. Let $\sigma(\overline{G}_r)$ denote an LSF schedule for (the tasks in) G_r when the part of the schedule corresponding to task Y is deleted[†]. Let L_r denote the length of $\sigma(\overline{G}_r)$. Finally our algorithm assigns the removed tasks in the back of $\sigma(\overline{G}_r)$ in the reverse order of their removal. We let $\sigma^+(S^*)$ be the schedule for the tasks in S^* obtained by the above procedure. The manner in which E_j and F_j were removed, implies that $\sigma^+(S^*)$ is a valid schedule for S^*, i.e., the precedence constraints for the tasks are not violated. The final schedule for $\mathcal{J}$ will then be the original LSF schedule for R^* followed by $\sigma^+(S^*)$. Figure 5 illustrates the final schedule for the tasks given in Fig. 1.

Lemma 4. *In the pruning process as described above, we will never encounter the situation where the only leaf task left is a single 2-unit task.*

Lemma 5. *The schedule $\sigma^+(S^*)$ is a minimum-length schedule for S^*.*

We can now summarize our algorithm as follows:

1. Construct LSF schedule $\sigma(\mathcal{J})$.
2. If $\sigma(\mathcal{J})$ is degenerate, go to Step 3; otherwise terminate the algorithm with $\sigma(\mathcal{J})$ being the final schedule.
3. Scan the LSF schedule $\sigma(\mathcal{J})$ backward and locate the task Y which finishes at t^*. If every task executed after t^* is a 2-unit task, then terminate the algorithm with $\sigma(\mathcal{J})$ being the final schedule; otherwise go to Step 4.
4. Prune the subtree rooted at Y by iteratively removing pairs of 2-unit leaf tasks farthest from Y. Terminate the pruning process as soon as a 1-unit task becomes a leaf.
5. Construct an LSF schedule $\sigma(G_r)$ for the pruned tree G_r. Find $\sigma(\overline{G}_r)$ from $\sigma(G_r)$ by deleting task Y. The final schedule will be the original schedule up to t^* followed by $\sigma(\overline{G}_r)$ which is then followed by the removed pairs of 2-unit tasks in the reverse order of their removal.

From the above lemmas and discussions, we have the following theorem.

Theorem 1. *The above algorithm generates a minimum-length schedule for a set $\mathcal{J}$ of tasks on two processors under the following assumptions:*

(1) there are two execution times, one being twice the other, and (2) the precedence relation among the tasks in $\mathcal{J}$ can be described by an out-tree.

If the LSF strategy is implemented by constructing a list of tasks and repeatedly scanning the list for assignment, the running time of the algorithm can be $O(n^2)$. We now briefly describe how our algorithm can be implemented to run in O(n log n) time. First by traversing the tree, we can compute W(T) and $|S(T)|$ for each task $T \in \mathcal{J}$. This can be done in O(n) time. Based on W(T) and $|S(T)|$, we will assign a priority $\rho(T)$ to each task T such that $\rho(T_i) > \rho(T_j)$ if either (1) $W(T_i) > W(T_j)$ or (2) $W(T_i) = W(T_j)$ and $|S(T_i)| > |S(T_j)|$. This can be done by using the simple function

---

[†] We need not reschedule Y since Y is not in S^*.

$\rho(T) = n \times W(T) + |S(T)|$. This step again takes linear time. The scheduling is performed by maintaining a heap of ready tasks. The heap is ordered by the priorities of the tasks. Initially, we schedule the root of the tree $G(\mathcal{T},E)$. Whenever a task finishes execution, we will put its immediate successors into the heap. We then remove the "largest" element from the heap and assign it to the processor. Since each task, except the root, is added to and deleted from the heap exactly once, the scheduling can be done in O(n log n) time. Thus it takes O(n log n) time to construct an LSF schedule. Step 2 can be done in constant time. Step 3 can be performed in O(n) time. Step 4 can be implemented to run in O(n log n) time by maintaining a heap of leaf tasks. The heap is ordered by the length of the path from Y to each task. Step 5 can be done in O(n log n) time. From the above discussion, we have the following theorem.

Theorem 2. The above algorithm can be implemented to run in O(n log n) time.

CONCLUSIONS

We have given an implementation of our algorithm which runs in O(n log n) time. One wonders if there is an alternative implementation which runs faster than O(n log n) time. For that matter, it is certainly conceivable that more efficient algorithms for the problem exist. In this connection, it is worth mentioning that our algorithm works if the LSF strategy is replaced by the Deepest-Subtree-First (DSF) strategy in which W(T) is replaced by, say D(T), where D(T) is the length of a longest path in $G(\mathcal{T},E)$ eminating from $T \in \mathcal{T}$. We chose LSF because its novelty may stimulate some thoughts.

We attempted to solve the same problem when the number of processors, $m > 2$. It seems that no simple modification of the algorithm given in this paper can lead to an optimal schedule even if $m = 3$. For example, consider the task system shown in Fig. 6 with its LSF schedule which is not an optimal schedule. An optimal schedule for the same task system is shown at the bottom of Fig. 6. No simple pruning technique seems to resolve the difficulties encountered here. It is possible that the problem of finding minimum-length schedules for tree precedence constrained task systems, with execution times 1 and 2, on $m > 2$ processors is NP-hard. However, it can be shown that in this case the result of Lemma 1 does hold.

The concepts involved in Lemmas 1 and 2 and their proofs may have applications beyond the scheduling problem considered here. In fact, Hakimi and Nakajima [7] state and prove a variation of Lemma 1, where m and the execution times are arbitrary, which immediately yields two classical results in the scheduling of tree precedence constrained tasks. These results were previously obtained by Hu [8] and Kaufman [9] with rather lengthy proofs.

REFERENCES

1. E. G. Coffman, Jr. and R. L. Graham, "Optimal sequencing for two processor systems," Acta Informatica, Vol. 1, No. 3, 1972, pp. 200-213.
2. E. G. Coffman, Jr. and J. Y-T. Leung, "Combinatorial analysis of an efficient algorithm for processor and storage allocation," SIAM J. on Computing, Vol. 8, No. 2, 1979, pp. 202-217.
3. E. G. Coffman, Jr., J. Y-T. Leung and D. W. Ting, "Bin-packing: maximizing the number of pieces packed," Acta Informatica, Vol. 9, 1978, pp. 263-271.
4. M. Fujii, T. Kasami and K. Ninomiya, "Optimal sequencing of two equivalent processors," SIAM J. on Applied Maths., Vol. 17, No. 3, 1969, pp. 784-789. Erratum, Vol. 20, No. 1, 1971, p. 141.

5. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, San Francisco, 1979.
6. E. Horowitz and S. J. Sahni, Fundamentals of Computer Algorithms, Computer Science Press, Potomac, MD., 1978.
7. S. L. Hakimi and K. Nakajima, "A unified proof for two classical results in scheduling theory," (in preparation).
8. T. C. Hu, "Parallel sequencing and assembly line problems," Operations Research, Vol. 9, No. 6, 1961, pp. 841-848.
9. M. T. Kaufman, "An almost-optimal algorithm for the assembly line scheduling problem," IEEE Transactions on Computers, C-23, No. 11, 1974, pp. 1169-1174.
10. J. Y-T. Leung, "On scheduling independent tasks with restricted execution times," Proc. of the 18th Annual Allerton Conference on Communication, Control and Computing, Oct. 1980.
11. J. D. Ullman, "NP-complete scheduling problems," J. of Computer and System Sciences, Vol. 10, 1975, pp. 384-393.
12. J. D. Ullman, "Complexity of sequencing problems," in Computer and Job-Shop Scheduling Theory, E. G. Coffman, Jr. (ed.), John Wiley and Sons, New York, 1976, pp. 139-164.

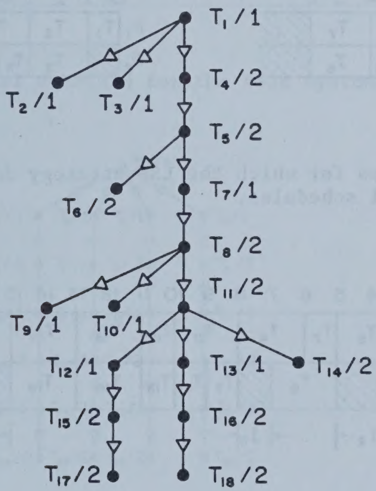


Fig. 1. A task system and the illustrations of some terms.

T_i/α means task T_i has executation time α. T_1 is the root of the tree; T_2, T_3, T_6, T_9, T_{10}, T_{14}, T_{17}, and T_{18} are leaves.

$S(T_1) = \{T_i \mid 1 \le i \le 18\}$, $W(T_1) = 28$, $W(T_{11}) = 14$, $S(T_{11}) = \{T_i \mid 11 \le i \le 18\}$, $D(T_1, T_{10}) = 9$, $D(T_{11}, T_{16}) = 5$, $|S(T_8)| = 11$.

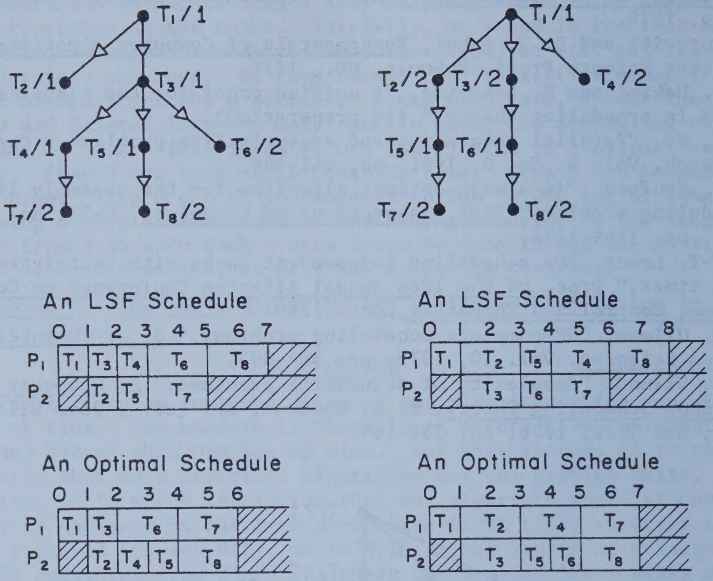

T₁/1
T₂/1
T₃/1
T₄/1
T₅/1
T₆/2
T₇/2
T₈/2
An LSF Schedule
0 1 2 3 4 5 6 7
P₁ T₁ T₃ T₄ T₆ T₈
P₂ T₂ T₅ T₇
An Optimal Schedule
0 1 2 3 4 5 6
P₁ T₁ T₃ T₆ T₇
P₂ T₂ T₄ T₅ T₈
T₁/1
T₂/2
T₃/2
T₄/2
T₅/1
T₆/1
T₇/2
T₈/2
An LSF Schedule
0 1 2 3 4 5 6 7 8
P₁ T₁ T₂ T₅ T₄ T₈
P₂ T₃ T₆ T₇
An Optimal Schedule
0 1 2 3 4 5 6 7
P₁ T₁ T₂ T₄ T₇
P₂ T₃ T₅ T₆ T₈

Fig. 2. Examples for which the LSF strategy does not yield optimal schedules.

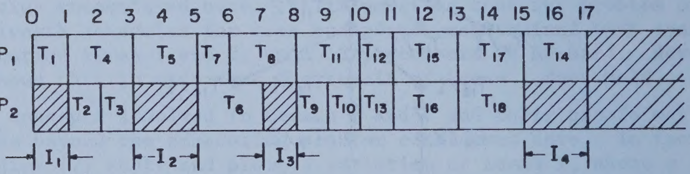

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
P₁ T₁ T₄ T₅ T₇ T₈ T₁₁ T₁₂ T₁₅ T₁₇ T₁₄
P₂ T₂ T₃ T₆ T₉ T₁₀ T₁₃ T₁₆ T₁₈
I₁
I₂
I₃
I₄

Fig. 3. An LSF schedule for the task system in Fig. 1 which illustrates certain concepts.

I_1, I_2 and I_3 are internal idle intervals; I_4 is the external idle interval.

$\sigma_1(\mathcal{T}) = \{T_1\}$, $\sigma_2(\mathcal{T}) = \{T_2, T_3, T_4, T_5\}$, $\sigma_3(\mathcal{T}) = \{T_6, T_7, T_8\}$, and $\sigma_4(\mathcal{T}) = \{T_i \mid 9 \le i \le 18\}$.

$s(T_1) = 0$ and $f(T_1) = 1$; $s(T_{15}) = 11$ and $f(T_{15}) = 13$. $f^* = 17$.

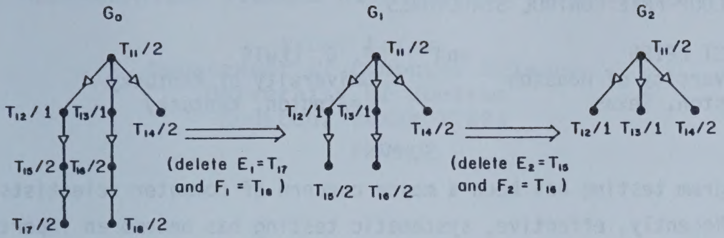

G₀
T₁₁/2
T₁₂/1
T₁₃/1
T₁₄/2
T₁₅/2
T₁₆/2
T₁₇/2
T₁₈/2
(delete E₁ = T₁₇ and F₁ = T₁₈)
G₁
T₁₁/2
T₁₂/1
T₁₃/1
T₁₄/2
T₁₅/2
T₁₆/2
(delete E₂ = T₁₅ and F₂ = T₁₆)
G₂
T₁₁/2
T₁₂/1
T₁₃/1
T₁₄/2

Fig. 4. Illustrating the pruning process.

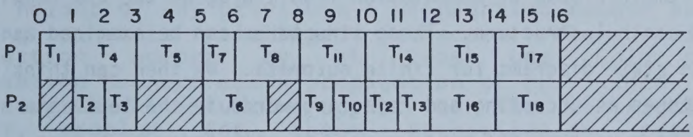

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
P₁
T₁
T₄
T₅
T₇
T₈
T₁₁
T₁₄
T₁₅
T₁₇
P₂
T₂
T₃
T₆
T₉
T₁₀
T₁₂
T₁₃
T₁₆
T₁₈

Fig. 5. The final schedule for the task system in Fig. 1.

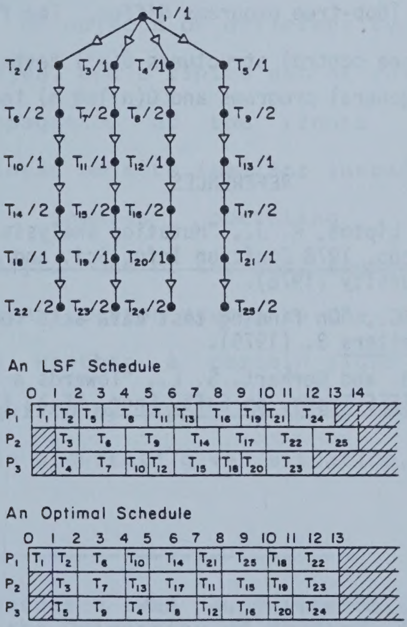

T₁/1
T₂/1
T₃/1
T₄/1
T₅/1
T₆/2
T₇/2
T₈/2
T₉/2
T₁₀/1
T₁₁/1
T₁₂/1
T₁₃/1
T₁₄/2
T₁₅/2
T₁₆/2
T₁₇/2
T₁₈/1
T₁₉/1
T₂₀/1
T₂₁/1
T₂₂/2
T₂₃/2
T₂₄/2
T₂₅/2
An LSF Schedule
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
P₁
T₁
T₂
T₅
T₈
T₁₁
T₁₃
T₁₆
T₁₉
T₂₁
T₂₄
P₂
T₃
T₆
T₉
T₁₄
T₁₇
T₂₂
T₂₅
P₃
T₄
T₇
T₁₀
T₁₂
T₁₅
T₁₈
T₂₀
T₂₃
An Optimal Schedule
0 1 2 3 4 5 6 7 8 9 10 11 12 13
P₁
T₁
T₂
T₆
T₁₀
T₁₄
T₂₁
T₂₅
T₁₈
T₂₂
P₂
T₃
T₇
T₁₃
T₁₇
T₁₁
T₁₅
T₁₉
T₂₃
P₃
T₅
T₉
T₄
T₈
T₁₂
T₁₆
T₂₀
T₂₄

Fig. 6. An example for 3-processor scheduling.

TESTING LOOP-FREE CONTROL STRUCTURES

ERNST LEISS and F. D. LEWIS
University of Houston
Houston, Texas

University of Kentucky
Lexington, Kentucky

SUMMARY

Program testing has been a major concern of computer scientists for years. Recently, effective, systematic testing has become an important research topic. Also, some of testing's theoretical aspects have been considered lately [1,2,3]. The work described here deals with the complexity of testing loop-free programs.

If we remove blocks of computation from a program the end result is a tree-shaped control structure. These structures can be examined as flowcharts or as state diagrams for finite automata. We then can think of these structures as accepting and rejecting words in the test data set. In order to be sure that the program accepts all words which must be accepted we need at least n/2 words in our test data set for a structure of n nodes. To ensure that the program being tested accepts no additional words, we find that the cases for line programs (transitions into a node's subtree) and general loop-free programs differ. The final result is:

THEOREM. For loop-free control structures $O(n^2)$ tests are needed to distinguish between general programs and $O(n \log n)$ tests are required for line programs.

REFERENCES

[1] Budd, T. A. and Lipton, R. J., "Mutation analysis of decision table programs", Proc. 1978 Conf. on Info. Sci. and Sys., The Johns Hopkins University (1978).

[2] Cherniavsky, J. C., "On finding test data sets for loop-free programs", Inf. Proc. Letters 8, (1979).

[3] Goodenough, J. B. and Gerhart, S. L., "Towards a theory of test data selection", IEEE Trans. on Soft. Engr. SE-1, (1975).

AUTHORIZATION SYSTEMS WITH BOUNDED PROPAGATION *

Ernst Leiss
Department of Computer Science
University of Houston
Houston, Texas 77004

Abstract

We discuss authorization systems where the length of the longest chain of grants of privileges is bounded by a fixed number. This is enforced by assigning to every holder of any privilege a propagation number such that passing on this privilege decreases the recipient's propagation number by at least one. Only positive propagation numbers allow propagation of privileges. We show that despite the intuitive appeal of the restriction the safety problem for these authorization systems is undecidable.

An authorization (or protection) system consists of objects of different types, for example users and data files, and a finite set of rules which govern the (legal) propagation of the rights or privileges pertaining to these objects (see for instance [2,5,7,9,4]; more oriented towards operating systems are [4,7,8,15,16,18,21]). One of the most fundamental questions concerning any authorization system is the safety problem; briefly it asks whether a certain right to a particular object can be acquired by applying some sequence of the rules, starting in a certain given situation.

---

*) This research was supported in part by a summer grant (RIG) from the University of Houston, Central Campus.

AUTHORIZATION SYSTEMS WITH BOUNDED PROPAGATION

Most of the work concerning authorization systems falls into one of two categories. One is well studied, with a clearly defined theory, and with well founded and sound theoretical results. Unfortunately, the authorization systems in this class are of little practical importance, either because the safety problem (and consequently also the more general confinement problem) is undecidable or cannot be decided in most (practical) situations (e.g. [6,7,13]), or else because the restrictions on the rules governing the propagation of privileges render the systems not very useful for real-world applications (e.g. [1,10,11,12,19,20]). The other class of authorization systems is that of actually implemented and used ones (e.g. [5,3,18,21]). Their disadvantage is that they suffer from a lack of theory (sound assertions about their behavior).

In [17], M. Rabin advocated a precise differentiation between the right to do something, the right to grant this right, the right to grant the right to grant this right, etc.. In this note we define a simple method which implements this idea, namely we bound the length of the longest chain of grants of the same right to any given object by a fixed number M. We believe that this assumption is almost always met for a reasonably small M. Furthermore, this method is a step towards guaranteeing that rights are not allowed to grow when they are passed on; this is known as the principle of attenuation and is discussed in [14].

More specifically, we propose a method based on the following observations. (We assume that there is a finite fixed number of rights.) Whenever an object C wants to give another object D some right r (to the object Z), C transmits to D the pair (r,m) with r being the right granted and m being an integer between 0 and M. This integer m is called the propagation number and is chosen by C; the non-negative integer M is a system constant. The significance of the propagation number m is as follows. Suppose A holds some right s to some object Y with propagation number n, and assume that A wants to grant B the right s (to Y). Then A will send B the pair (s,n') where n' must be an integer (chosen by A) between 0 and n-1. In other words, if B receives right s from A, B's propagation number will be at least one less than A's. Clearly, the propagation number depends on A, s, and Y. This implies that the length of the longest chain of grants of some right concerning any given object will always be bounded by M.

A slightly more restrictive version of this model proposes that any object which grants a right to any other object will thereby decrease its corresponding propagation number. This restriction implies that given any right to any object there will be at most M objects other than its creator holding this right. Note that propagation number 0 means that the user can exercise the right but cannot pass it on.

Since intuitively both methods impose a severe restriction on the model one would expect that at least the safety problem is decidable. However, this is not so:

Theorem: The safety problem for authorization systems with bounded propagation is undecidable.

The idea of the proof is to modify the original HRU proof ([7]) which requires M to be unbounded in order to be able to simulate an arbitrary Turing machine with a fixed finite M. In fact, a careful analysis shows that even with $M=0$ the safety problem is not decidable.

The implication of this result is quite clear: While it is certainly important for an accurate model of the real world to include the distinction between a right and the right to grant this right, restrictions based on limiting consecutive grants, and by extension restrictions based on the principle of attenuation, do not necessarily imply the solvability of the safety problem. This is another indication that the solvability of the safety problem can not be easily enforced by general restrictions on the granting mechanisms. (It is however guaranteed if the number of subjects is fixed; see [13].)

References

[1] BISHOP, M., SNYDER, L. The Transfer of Information and Authority in a Protection System
Department of Computer Science, Technical Report #166, Yale University, July 1979

[2] DENNING, D. E., DENNING, P. J. Data Security
ACM Computing Surveys, Vol. 11, No. 3, September 1979, 227-250

[3] FAGIN, R. On an Authorization Mechanism
ACM ToDS 3,3, 310-319, Sept. 1978

[4] GRAHAM, G. S., DENNING, P. J. Protection, Principles and Practice
Proc. 1972 AFIPS Spring Joint Computer Conference, Vol. 40, AFIPS Press, Montvale, N.J., 417-429, 1972

[5] GRIFFITHS, P. P., WADE, B. W. An Authorization Mechanism for a Relational Database System
ACM ToDS 1,3, 242-255, Sept. 1976

[6] HARRISON, M. A., RUZZO, W. L. Monotonic Protection Systems
pp. 337-365 in [22]

[7] HARRISON, M. A., RUZZO, W. L., ULLMAN, J. D. Protection in Operating Systems
CACM 19,8, 461-471, Aug. 1976

[8] JONES, A. K. Protection in Programmed Systems
Ph. D. Thesis, Carnegie-Mellon University, 1973

[9] JONES, A. K. Protection Mechanisms and the Enforcement of Security Policies
pp. 229-251 in [23]

[10] JONES, A. K. Protection Mechanism Models: Their Usefulness
pp. 237-254 in [22]

[11] JONES, A. K., LIPTON, R. J., SNYDER, L. A Linear Time Algorithm for Deciding Security
Proc. 17th FOCS, 1976

[12] LIPTON, R. J., BUDD, T. A. On Classes of Protection Systems
pp. 281-298 in [22]

References

[13] LIPTON, R. J., SNYDER, L. On Synchronization and Security
pp. 367-385 in [22]

[14] MINSKY, N. The Principle of Attenuation of Privileges and its Ramifications
pp. 225-257 in [22]

[15] POPEK, G. J., FARBER, D. A. A Model for Verification of Data Security in Operating Systems
CACM 21,9, 737-749, Sept. 1978

[16] POPEK, G. J., KLINE, C. S. Issues in Kernel Design
Proc. AFIPS Nat. Computer Conf. Vol. 47, AFIPS Press, Montvale, N. J., 1079-1086, 1978

[17] RABIN, M. O. Comment on p. 277 in [22]

[18] SCHROEDER, M. D., CLARK, D. D., SALTZER, J. H. The MULTICS Kernel Design Project
Proc. Sixth Symp. on Operating System Principles, Operating Systems Rev. (ACM) 11,5, 43-56, Nov.1977

[19] SNYDER, L. Synthesis and Analysis of Protection Systems
Proc. Sixth Symp. on Operating Systems Principles, 1977

[20] SNYDER, L. Theft and Conspiracy in the Take-Grant Protection Model
Dept. of Computer Science, Technical Report #147, Yale University, Nov. 1978

[21] STONEBRAKER, M., RUBINSTEIN, R. The INGRES Protection System
Proc. ACM National Conference 1976, 80-84, 1976

[22] DEMILLO, R. D., DOBKIN, D., JONES, A. K., LIPTON, R. J. (eds.) Foundations of Secure Computation
Academic Press, New York, 1978

[23] FLYNN et al. Operating Systems: An Advanced Course
Springer Lecture Notes in Computer Science, Vol. 60, Springer Verlag, New York, 1978

MARKED GRAPH TRANSFORMATIONS BY STEPWISE TRANSITION REFINEMENTS[*]

ICHIRO SUZUKI AND TADAO MURATA
Department of Information Engineering
University of Illinois at Chicago Circle
Chicago, Illinois 60680

ABSTRACT

This paper shows that several transformations of marked graphs reported earlier can be derived from a transformation method for Petri nets. The latter is a stepwise transformation of a transition into a subnet (or vice versa) such that liveness and safeness properties are preserved.

I. INTRODUCTION

Recently it has been shown [1,2] that $\rho(G)$, the number of equivalence classes of live and safe markings, is either invariant or computed by a formula when a certain transformation (reduction or expansion) is applied to a marked graph G. So far six types of such transformations are known and they are referred to as series, parallel, unique circuit, Y-V, separable graph, and unique path transformations. These transformations are useful for both analysis and synthesis of marked graphs. They can be used to synthesize concurrent systems modeled by marked graphs, with the following prescribable properties: liveness, safeness, performance, and resources (the maximum wieghted sum of tokens, the number of states, etc.) [3,4].

While the above transformations are applicable only to marked graphs — a subclass of Petri nets, there have also been reported transformations applicable to live and/or safe Petri nets [5] and to more general Petri nets [6]. These Petri net transformations are a method to refine (or abstract) a transition into a subnet (vice versa) step by step, and are sometimes called stepwise refinements (abstractions).

The marked graph transformations can deal with more properties (such as $\rho(G)$, performance and resource requirements) than the Petri net transformations can, but the former should be special cases of the latter as far as liveness and safeness properties are concerned. This paper shows that the marked graph transformations can indeed be derived from the Petri net transformations. Thus, it provides another way of proving that the marked graph transformations preserve liveness and safeness.

---

[*] This work was supported by NSF Grant ENG 78-05933.

II. STEPWISE TRANSITION REFINEMENTS

This section briefly reviews the Petri net transformation given by Valette [5], which will be used to explain the six types of marked graphs transformations in the next section. A more general transformation [6] could be used here but the transformation by Valette will suffice for the present purpose since we are concerned with live and safe Petri nets.

A stepwise transition refinement is a method for substituting a transition t_r in a Petri net N by another Petri net N' to form a refined Petri net N". A Petri net N' is called a block when a pair of distinct transitions t_{in} and t_{out} in N' are fixed. A block to be substituted for a transition t_r must satisfy certain conditions (well-formedness defined below). For the purpose of testing these conditions, we form the associated Petri net $\hat{N}'$of block N' by adding a place p_0 called the idle place with an arc from p_0 to t_{in} and another arc from t_{out} to p_0 in N'. Let $\hat{M}_0$ be an initial marking of $\hat{N}'$ such that t_{in} is the only transition enabled at $\hat{M}_0$. Then, a block N' is said to be well-formed if (1) $\hat{N}'$ is live for $\hat{M}_0$ and (2) among all the markings reachable from $\hat{M}_0$, $\hat{M}_0$ is the only marking such that the idle place is not empty.

In the associated Petri net $\hat{N}'$ of a well-formed block N' with the initial marking $\hat{M}_0$, all firing sequences start from t_{in}, once t_{in} fires it cannot fire second time until after t_{out} fires, and only when t_{out} fires the marking goes back to the initial marking $\hat{M}_0$. Thus, it can be seen intuitively that a well-formed block may be substituted for a transition. However, not every transition can be replaced by a well-formed block (to preserve liveness). A transition t_r to be replaced must be not two-enabled (i.e. it cannot fire twice consecutively) as stated in the following theorem due to Valette [5]. (Note that a transition, which is not (k+1)-enabled for any positive integer k, is replaceable in the method of Suzuki and Murata [6].)

Theorem 1: Let t_r be a transition which is not two-enabled at any reachable marking in a Petri net N. Let N" be the Petri net obtained from N by substituting a well-formed block N' for t_r. Then, we have:

(a) The Petri net N" is bounded iff the Petri net N is bounded.

(b) The Petri net N" is safe iff the Petri net N and the block N' are safe.

(c) The Petri net N" is live iff the Petri net N is live.

III. TRANSFORMATIONS OF LIVE AND SAFE MARKED GRAPHS

In this section, the six types of marked graph transformations given in [1,2] will be interpreted in terms of Petri net transformation described in Section II. Note that since the marked graph transformations considered here are transformations among live and safe marked graphs, each transition to be replaced in the following is not two-enabled and satisfies the hypothesis in Theorem 1. Throughout this paper, transformation means both expansion (refinement) and reduction (abstraction), and the latter is the reverse operation of the former. Note that Theorem 1 applies to both expansion and reduction.

(i) Series Transformation: Series expansion adds a place e' in series with place e as is shown in Fig. 1(a) and (b), which can be regarded as the refinement of t_r by block N' as is illustrated in the same figure. It is easily seen that the block N' consisting of t_{in}, e' and t_{out} is well-formed (the associated Petri net $\hat{N}'$ is shown in Fig. 1(c)). Thus, by Theorem 1, N" is live and safe iff N is live and safe, and both series expansion and reduction preserve liveness and safeness.

(ii) Parallel transformation: Parallel expansion adds a place e' in parallel with a place e as is shown in Fig. 2(a) and (c). This transformation is regarded as a combination of the two Petri net transformations shown in Fig. 2(a) to (b) and from Fig. 2(b) to (c), where the former is an abstraction to a transition t_r and the latter is a refinement of t_r. It is easily seen that the blocks are well-formed in both cases. Thus, by Theorem 1, N" is live and safe iff N is live and safe, and parallel expansion and reduction preserve liveness and safeness.

(iii) Y-V Transformation: Fig. 3 illustrates Y-V expansion. It is easily seen that this expansion is the same type of refinement of a transition t_r as the series expansion. Thus, the same argument as in (i) applies.

(iv) Unique Circuit Transformation: Where there exists a unique directed path p_{12} from a transition t_1 to another transition t_2, addition of a place p together with an arc from t_2 to p and another arc from p to t_1 creates a unqiue directed circuit and is called the unique circuit expansion. Before adding place p, each place on p_{12} has been made token-free. In this case a given Petri net with a unique directed path p_{12} from t_1 to t_2 can be regarded as a well-formed block with respect to t_1 and t_2. This expansion can be viewed as the refinement of transition t_r in the live and safe Petri net shown in Fig. 4(a) by a given Petri net, or as the addition of the idle place p to a given Petri net as is illustrated in Fig. 4. Thus, by Theorem 1 unique circuit transformation preserves liveness and safeness.

(v) Separable Graph Transformation: Separable graph expansion merges a transition t_1 of a live and safe marked graph N_1 and a transition t_2 of another live and safe marked graph N_2 to create a separable graph N". This expansion is achieved by applying the Petri net transformation three times as is illustrated in Fig. 5. First, t_1 in N_1 is refined into $\{t_{in}, p_1, t_{out}\}$ to form N_1' as shown in Fig. 5(b). Second t_2 in N_2 is refined by the well-formed block N_1' to obtain N_3 as shown in Fig. 5(c). Finally, the well-formed block $\{t_{in}, p_1, t_{out}\}$ in N_3 is reduced to a single transition t_3 as is shown in Fig. 5(d).

(vi) Unique Path Transformation: Fig. 6 illustrates unique path expansion where p_{12} (p_{34}) is a token-free unique directed path from t_1 (t_3) to t_2 (t_4) in a live and safe marked graph N_1 (N_2), t_1 and t_3 (t_2 and t_4) are merged to a single transition t_5 (t_6), and a new place p from t_6 to t_5 is added in the newly merged marked graph N". Since N_1 (N_2) is well-formed with respect to t_1 and t_2 (t_3 and t_4), N' shown in Fig. 7 is well-formed with respect to t_5 and t_6 (note that each place on p_{12} and p_{34} has been made token-free). Then this expansion can be regarded as the addition of the idle place p to the well-formed block N' or as the refinement of transition t_r in the live and safe Petri net N by N' as is illustrated in Fig. 8. Thus, unique path transformation preserves liveness and safeness as well as the other five transformations do.

REFERENCES

[1] T. Murata and J.Y. Koh, "Reduction and expansion of live and safe marked graphs", IEEE Trans. on Circuits and Systems, Vol. CAS-27, No. 1, pp. 68-70, January, 1980.

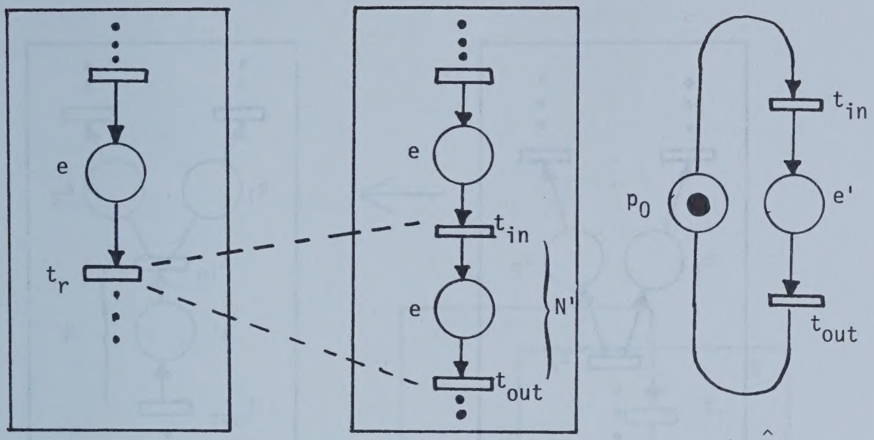
[2] R. Johnsonbaugh, M.C. Kao and T. Murata, "Additional transformations of live and safe marked graphs", Procs. of the 17th Allerton Conference on Communications, Control, and Computing, pp. 387-396, October, 1979.

[3] T. Murata, "Synthesis of marked graph computation models for prescribed resources and performance", Procs. of COMPSAC '79, pp. 807-812, November, 1979, a revised version to appear in the IEEE Trans. on Software Engieering, November, 1980.

[4] R. Johnsonbaugh and T. Murata, "Analysis of resource requirements in marked graph computation models", Procs. of the 1980 International Symposium on Circuits and Systems, pp. 342-345, April, 1980.

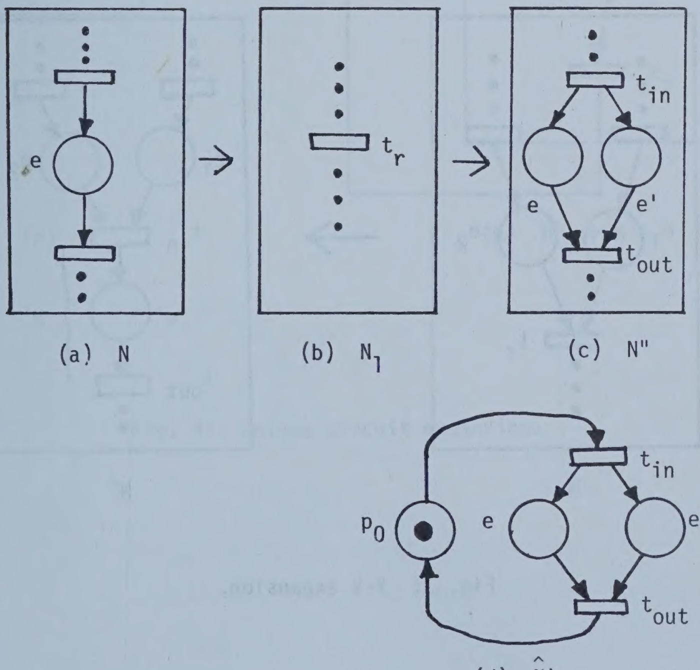
[5] R. Valette, "Analysis of Petri nets by stepwise refinements", J. of Computer and System Sciences, Vol. 18, No. 1, pp. 34-46, February, 1979.

[6] I. Suzuki and T. Murata, "A method for hierarchically representing large scale Petri nets", Procs. of ICCC/80, October, 1980.


e
t_r
e
t_{in}
e
N'
t_{out}
t_{in}
p_0
e'
t_{out}

(a) N (b) N" (c) $\hat{N}'$

Fig. 1: Series expansion.


e
t_r
t_{in}
e
e'
t_{out}
t_{in}
p_0
e
e'
t_{out}

(a) N (b) N_1 (c) N"

(d) $\hat{N}'$

Fig. 2: Parallel expansion.

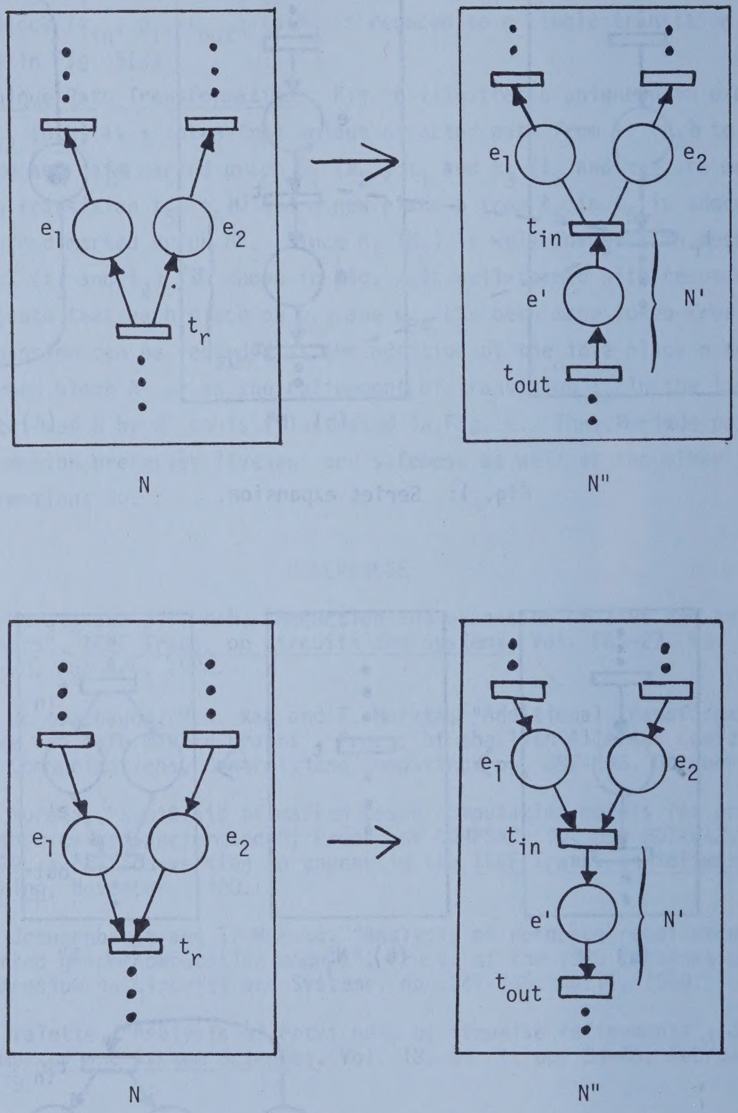
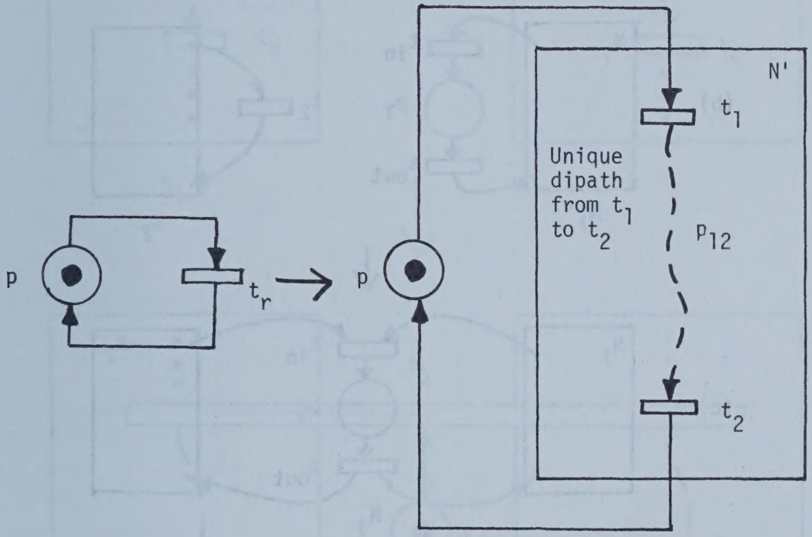

e1
e2
tr
N
e1
e2
tin
e'
N'
tout
N"
e1
e2
tr
N
e1
e2
tin
e'
N'
tout
N"

Fig. 3: Y-V expansion.


p
t_r
p
N'
t_1
Unique dipath from t_1 to t_2
p_{12}
t_2

(a) N

(b) N" or $\hat{N}$'

Fig. 4: Unique circuit expansion.

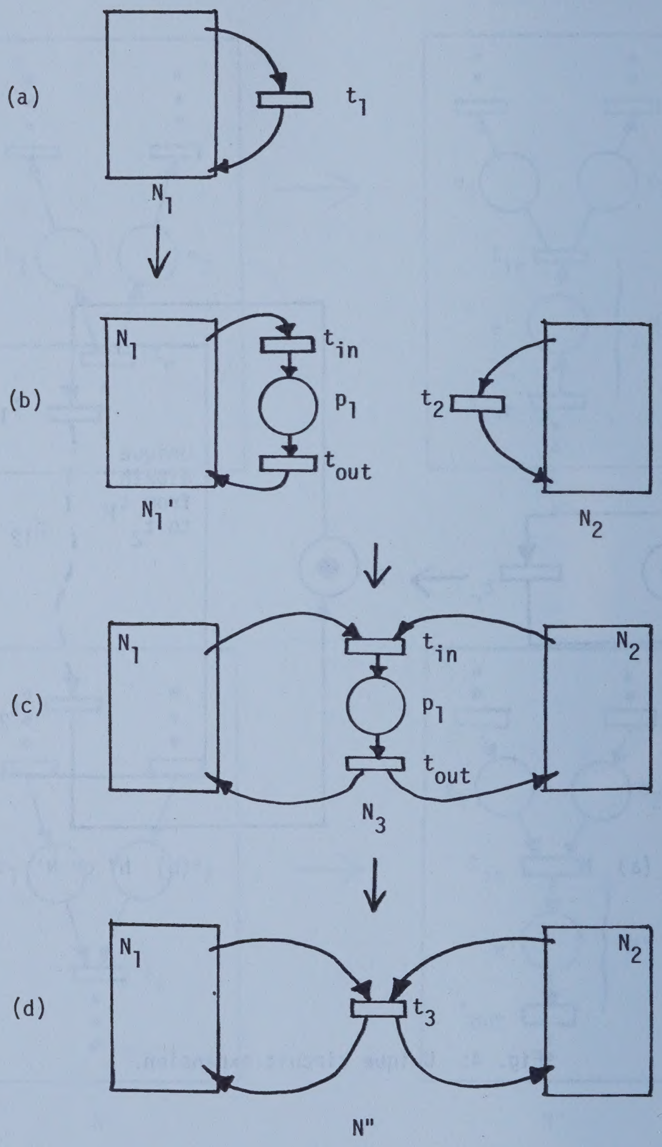
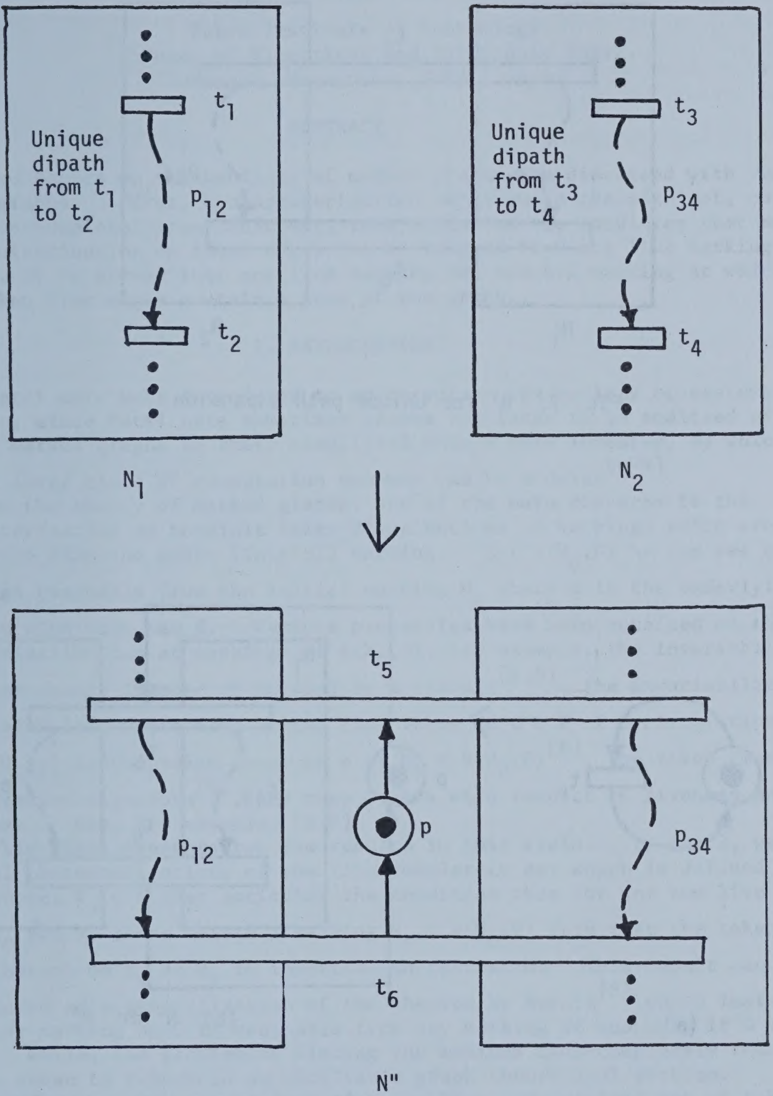

(a)
t_1
N_1
(b)
N_1
t_in
p_1
t_out
N_1'
t_2
N_2
(c)
N_1
t_in
p_1
t_out
N_2
N_3
(d)
N_1
N_2
t_3
N"

Fig. 5: Separable graph expansion.



t1
Unique dipath from t1 to t2
p12
t2
N1
t3
Unique dipath from t3 to t4
p34
t4
N2
t5
p
p12
p34
t6
N"

Fig. 6: Unique path expansion.

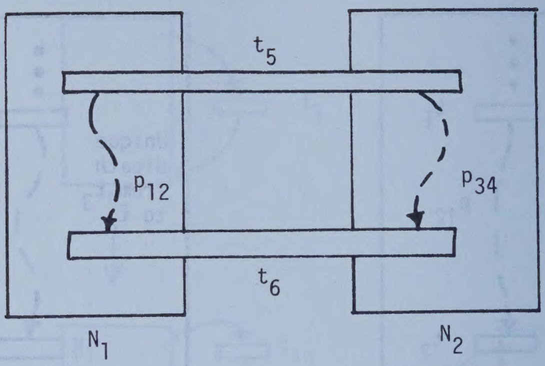
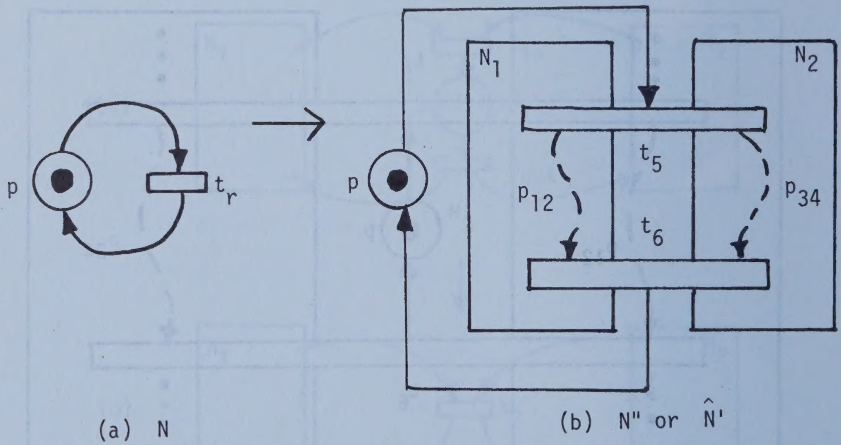

t_5
p_12
p_34
t_6
N_1
N_2

Fig. 7: N' for unique path expansion


p
t_r
p
N_1
N_2
t_5
t_6
p_12
p_34

(a) N

(b) N" or $\hat{N}'$

Fig. 8: Another interpretation of unique path expansion.

SUBGRAPH REACHABILITY OF MARKED GRAPHS

YOJI KAJITANI and TOSHIO KAWAKAMI

Tokyo Institute of Technology
Dept. of Electrical and Electronic Engrg.
Ookayama, Meguro-ku, Tokyo, Japan

ABSTRACT

Two topics on reachability of marked graphs are discussed with respect to subgraphs. First, a characterization is given to the edge set, called the live-completely reachable set, that satisfies the condition that any token distribution on these edges can be reached from any live marking. Second, it is proved that any live marking can reach a marking at which the the token free edges contain a tree of the graph.

1. INTRODUCTION

Petri nets were introduced to represent a wide variety of systems[1-5]. However, since Petri nets sometimes become too large to be analized effectively, marked graphs as their simplified models were invented, by which a pretty large class of computation schemes can be modeled[6-9].

In the theory of marked graphs, one of the main concerns is the characterization of possible token distributions at markings which are reachable from the given (initial) marking. Let $R(M_0,G)$ be the set of all markings reachable from the initial marking M_0 where G is the underlying digraph with edge set E. Various properties have been obtained on the token distribution at markings of $R(M_0,G)$, for example, the invariability of token counts (number of tokens) in a circuit[6,8], the invariability of $\sum f(e)\cdot M_i(e)$ where f(e) is the flow value on $e \varepsilon E$ of a circulation flow f and $M_i(e)$ is the token count on e at $M_i \varepsilon R(M_0,G)$[6], the token count minimization algorithm[6], and many others with respect to liveness and safeness. See, for example, [8,9].

This paper presents two new results in this field. In Sec.2, we give several characterizations of the live-completely set which is defined as an edge subset $E_s \subset E$ that satisfies the condition that for any two live markings M_0 and M, there exists a marking $M_i \varepsilon R(M_0,G)$ such that the token distribution on E_s at M_i is identical to that at M. This result can be considered as a generalization of the theorem by Murata[8] which insists that any marking on G is reachable from any marking if and only if G is a tree. While, the problem of finding the maximum live-completely reachable set is shown to reduce to an unsolvable graph theoretical problem.

In Sec.3, it is proved that if M_0 is live, the maximal set of token free edges (edges without tokens) at markings that belong to $R(M_0,G)$ contains a tree of G. By this theorem, we can characterize $R(M_0,G)$ by taking such a tree and tokens placed only on its cotree. The result is proved as a corollary of a more general graph theoretical theorem, which is proved here, that in any positively weighted graph, there exists a tree such that the algebraic sum of weights of edges in each fundamental circuit is non-negative,

Considering the fact that in vector addition systems, the general reachability problem is reduced to the zero-reachability problem[10], the study of the distribution of token free edges must be one of the most important themes. And if we interpret marked graphs as plans of a concurrent processing including a number of machines, this bears direct relations to the schedule of repairing the machins.

This paper assumes the readers familiar with the knowledge on marked graphs as seen in [6,8].

2. REACHABILITY OF MARKINGS WITH RESPECT TO A SUBGRAPH

For a marking M and $E_s \subset E$, let $M(E_s)$ denote the token distribution on E_s at M, which may be represented by a vector as shown in EXAMPLE 2.

Let M_0 be a marking (the initial marking). For $E_s \subset E$ and a marking M, if there exists an $M_i \in R(M_0,G)$ such that $M(E_s) = M_i(E_s)$, M is said to be reachable from M_0 with respect to E_s. If $E_s = E$, this reduces to the usual definition of 'reachability'. If an arbitrary M is reachable from any M_0 with respect to E_s, E_s is said to be completely reachable. As a special case when only the live markings are considered, it is said a live-completely reachable set. It is known that the whole edge set E is completely reachable if and only if G is circuit free.[4] Theorem 1 in the following characterizes the live-completely reachable set.

Before the theorem, several definitions and a lemma are necessary.

For an edge e and a circuit C such that $e \in C$, $C_+(e)$ denotes the set of edges of C other than e whose direction is coincident with that of e in C. While $C_-(e)$ denotes the set of edges whose direction is opposite to that of e. Note that

$$C_+(e) \cup C_-(e) = C - \{e\}.$$

Similarly, for e and a cutset B such that $e \in B$, $B_+(e)$ and $B_-(e)$ are defined as the sets of edges of $B - \{e\}$ whose direction is coincident with and opposite to that of e, respectively, in B.

For $e \in E$, let w(e) be the weight of e (any real number assigned on e, for example, the token count on e). For a circuit C and $e \in C$, the algebraic sum w(C) of C along with e is

$$w(e) + \sum_{e \in C_+(e)} w(e) - \sum_{e \in C_-(e)} w(e) .$$

For a cutset B and $e \in B$, the algebraic sum of B along with e is similarly defined.

LEMMA 1:

Following three conditions for $E_s \subset E$ are equivalent.

(*) For any $e \in E_s$ and any circuit C that contains e,

$$(E-E_s) \cap C_+(e) \neq \emptyset \quad \text{and} \quad (E-E_s) \cap C_-(e) \neq \emptyset.$$

(**) For any $e \in E_s$, e is separable, or there exist two cutsets B_i and B_j such that

$$B_i \cap E_s = \{e\} \quad , \quad B_j \cap E_s = \{e\},$$

and

$$B_{i-}(e) = \emptyset \quad , \quad B_{j+}(e) = \emptyset.$$

(***) Let E_q be the set of edges of strongly connected components of G. Then,

(1) $E_s \cap E_q = \emptyset$,

(2) E_s contains no circuits in $G(E_q:)$ where $G(E_*:)$ is the graph obtained from G by contracting the edges of E_*, and

(3) $G(E_q \cup E_s:)$ is acyclic.

Proof is omitted.

THEOREM 1:

For any graph G, $E_s \subset E$ is live-completely reachable if and only if E_s satisfies the condition in LEMMA 1.

PROOF: Suppose E_s satisfies (**). Let M_0 be the initial live marking arbitrarily given and M be a marking where $M(E_s)$ is assigned arbitrarily. Then the proof of sufficiency part of the theorem is to show that $M(E-E_s)$ can be determined such that $M \varepsilon R(M_0,G)$.

Let M' be an auxiliary marking on G which allows even negative token counts on edges. It is constructed as follows. First, put M' as

$$M'(E) = 0 \text{ (zero vector).}$$

It is obvious that M' satisfies KVL, i.e. the algebraic sum of token counts of any circuit along with any of its edges is zero.

For $e \varepsilon E_s$, let

$$\delta(e) = M(e) - M_0(e).$$

If $\delta(e) \geq 0$, find B_i as in (**) and add $\delta(e)$ to every edge of B_i. The resultant M' again satisfies KVL. (NOTE: The weight distribution in a digraph satisfies KVL if and only if it is given as a linear combination of directed weighted cutset vectors.) While, if $\delta(e) < 0$, find B_j as in (**) and add $\delta(e)$ to e and $-\delta(e)$ to every edge of $B_j - \{e\}$. Again the resultant M' satisfies KVL. Continue this operation to all edges of E_s to obtain M', in which KVL is satisfied and each $e \varepsilon E_s$ has the weight $\delta(e)$ which may be negative and $e \varepsilon E-E_s$ has non-negative weight.

Construct a marking $M'' = M_0 + M'$ (vector addition). Then

$$M''(e) = M(e) \quad \text{if} \quad e \varepsilon E_s, \text{ and}$$

$$M''(e) \geq 0 \quad \text{if} \quad e \varepsilon E-E_s.$$

Thus M" is a marking. On the other hand, it is known[3] that any marking is reachable from a live marking if and only if their difference of token distribution satisfies KVL. Therefore, M" is the required marking.

Next, suppose E_s does not satisfy (*), i.e. there exists a circuit C such that

CASE 1: $(E-E_s) \cap C_+(e) = \emptyset$,

or

CASE 2: $(E-E_s) \cap C_-(e) = \emptyset$.

In CASE 1, consider the situation that $M_0(e)=K>0$, $M_0(E-\{e\})=0$ and $M(E_s)=0$. Then, the algebraic sum of C along with e at M_0 is K>0 and that at M, whatever we define $M(E-E_s)$, is non-positive. Because the algebraic sum of a circuit is invariant at markings of $R(M_0,G)$, M cannot belong to $R(M_0,G)$.

In CASE 2, consider the situation that $M_0(E)=0$, $M(e)=K>0$, $M(E_s-\{e\})=0$.

The algebraic sum of C along with e at M_0 is 0 while that at M is K or more whatever we define $M(E-E_s)$. Thus $M \notin R(M_0,G)$.

Thus in any case, E_s is not a live-completely reachable set.

Q.E.D.

We have by LEMMA 1 three characterizations of live-completely reachable set. As for each edge, we have the following characterization.

An edge e is said to be transitive if there exists a circuit C which includes e and $C_+(e)=\emptyset$.

COROLLARY 1:

An edge e does not belong to any live-completely reachable set if and only if e belongs to a strongly connected component or a transitive edge.

Proof is easy and omitted here.

Important is that it is not always true that the set of edges each of which does not satisfy the condition of the corollary forms a live-completly reachable set. See the following example.

EXAMPLE 1:

Suppose G shown in Fig.1 is the underlying graph. The edges e_4, e_5 and e_6 are the transitive edges. But the rest $\{e_1,e_2,e_3\}$ is not a live-completely reachable set. In fact, $\{e_1\}$ and $\{e_2,e_3\}$ are the maximal sets of live-completely reachable sets.

END

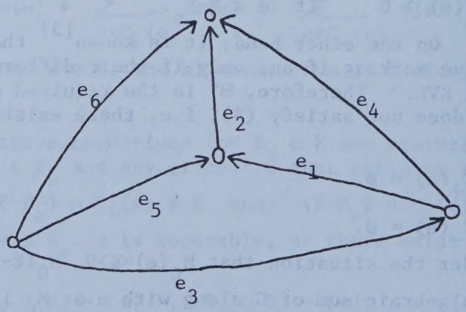
As seen in this example, even the maximal live-completely reachable set is not unique in cardinality. Thus, there is the problem to find the live-completely reachable set of maximum cardinality. This is, considering condition (***) of LEMMA 1, essentially stated as follows.

(P) In a given acyclic digraph G, find a maximum $E_s \subset E$ such that E_s is circuit free and $G(E_s:)$ remains acyclic.

Note that the dual problem to this is as follows.

(P^d) In a given strongly connected digraph G, find a maximum $E_s \subset E$ such that E_s is cutset free and $G(:E_s)$ remains strongly connected, where $G(:E_*)$ is the graph obtained from G by deleting the edges of E_*.

We believe that those two problems both belong to the NP-complete class.



e_4, e_5, e_6 : transitive

$\{e_1\}$, $\{e_2,e_3\}$: maximally live-completely reachable

Fig.1 Underlying graph G which is acyclic

3. TOKEN FREE EDGES AT MARKINGS OF $R(M_0,G)$

This section is devoted to the consideration of the maximal set of edges without tokens (token free edges) at markings that belong to $R(M_0,G)$ where M_0 is an arbitrary live marking.

First, some graph theoretical lemma will be presented.

Let G be an edge-weighted digraph. For a tree t of G, let $C(t,e)$ ($e \notin t$) be the fundamental circuit including e with respect to t. Let $B(t,e)$ ($e \in t$) be the fundamental cutset including e with respect to t.

The following lemma is, though we believe it a new one, so basic that it may be found in some previous works.

LEMMA 2:

Let G be a digraph whose edge-weight is non-negative. Then:

(C+) There exists a tree t such that for any $e \notin t$, the algebraic sum of $C(t,e)$ along with e is non-negative.

(B+) There exists a tree t such that for any $e \in t$, the algebraic sum of $B(t,e)$ along with e is non-negative.

PROOF of (C+): Find an edge set E_t by the following algorithm.

```
                    NON-NEGATIVE-CIRCUIT TREE
STEP 1: Let $E_t = \emptyset$.
STEP 2: Chose any cutset B satisfying
            $B \cap E_t = \emptyset$.
        If such B does not exist, END.
STEP 3: Find $e \in B$ such that $w(e)$ is minimum of the members of B, and let
            $E_t = E_t \cup \{e\}$.
STEP 4: Change the weights of the edges of G to obtain the graph G' as
        follows.
            If $f \in B_+(e) \cup \{e\}$, $w(f) = w(f) - w(e)$,
            if $f \in B_-(e)$, $w(f) = w(f) + w(e)$, and
            if $f \notin B$, $w(f) = w(f)$.
        Put G = G' and return to Step 2.
                                                                   END
```

Note that in the resultant graph G', E_t is a tree whose edges are all weight 0 and that the weights of edges not in E_t is non-negative. Thus, in G', E_t is a tree as described in (C+). Further, it is easy to see that the weight change in STEP 4 does not affect the algebraic sum of each circuit. Then, E_t is the tree of G as described in (C+).

The proof of (B+) can dually be made.

Q.E.D.

Once this lemma is proved, the following theorem is obvious.

The set of token free edges at a marking M is denoted as $E_0(M)$.

THEOREM 2:

Suppose M_0 is live. Then, there exists a marking $M_i \in R(M_0,G)$ such that $E_0(M_i)$ contains a tree.

PROOF: Consider G as a digraph whose edge-weight is defined by $M_0(E)$. Obtain G' from G by applying the algorithm NON-NEGATIVE-CIRCUIT TREE. The edge-weight in G' is the non-negative integer. Let M_i be the marking considering these edge-weights being the token counts on the edges.

It is obvious that the operation in STEP 4 is such that the difference of edge-weights of G and G' satisfies KVL. Then, since M_0 is live, $M_i \in$

ε R(M_0,G) and M_i(E) contains a tree whose token count is 0.

Q.E.D.

We can get the dual theorem corresponding to (B+). But its meaning is not clear in the theory of marked graphs.

Unfortunately, there exists a tree which satisfies (B+) but cannot be obtained by the algorithm NON-NEGATIVE-CIRCUIT TREE as shown in the following example. Thus, the complete characterization of the trees satisfying (B+) remains as a problem.

EXAMPLE 2:

In Fig.2, $M_0(E) = (w(e_1)\ w(e_2)\ \ldots\ldots\ w(e_8)) = (4\ 6\ 4\ 6\ 3\ 3\ 3\ 3)$ is shown. If we fire each vertex the number shown in the figure of times, we get a token free tree $\{e_1,e_2,e_3,e_4\}$. Note that this tree is not the one obtainable by NON-NEGATIVE-CIRCUIT TREE, because any cutset contains one of e_5, e_6, e_7 and e_8 as a minimum-weighted edge.

END

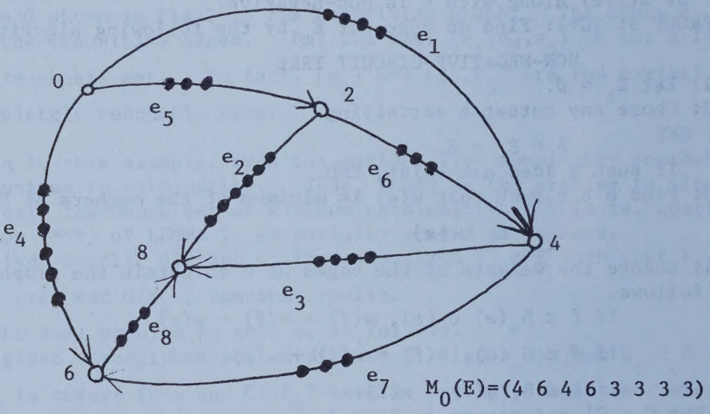


Fig.2 Live marking M_0

4. CONCLUSION

Reachability of markings on subgraphs was studied, and two main results, one is on live-completely reachable set and the other on the token free edges, were obtained. They are considered as generalizations of the results on complete reachability (controllability) and zero reachability, respectively.

There are many problems around the subgraph reachability. In fact, this paper itself contains in some of its essential parts several unsolved problems which are stated in the last paragraphs of Sec.2 and 3.

ACKNOWLEDGEMENT: The authors wish to express their thanks to Dr. S. Ueno of Tokyo Institute of Technology for his valuable suggestions, and also to Prof. T. Murata of Illinois Univ. for his kind and warmful advices.

REFERENCES

[1] C. A. Petri:"Kommunikation mit Automaten,"Schriften des Rheinisch-Westfalishen Institutes fur Instrumentelle Mathematik an der Universutat Bonn, Heft.2, Bonn, 1962.

[2] R. M. Karp and R. E. Miller:"Properties of a model for parallel computation; Determinancy, termination, queueing," SIAM J. App. Math. 14,,1390-1411 (1966).

[3] R. M. Karp and R. E. Miller:"Parallel program schemata," J. Comp. and Syst. Sci., 3, 4, 167-195 (May 1969).

[4] R. Reiter:"Scheduling parallel computations," J. of ACM, 15, 4, 590-599 (Oct. 1968).

[5] Y. E. Lien:"Termination properties of generalized Petri nets," SIAM J. Comput. 5, 2, 251-265 (June 1976).

[6] F. Commoner, A. W. Holt, S. Even and A. Penueli:"Marked directed graphs," J. Comput. and Syst. Sci. 5, 511-532 (Oct. 1971).

[7] A. W. Holt and F. Commoner:"Events and condition," Applied Data Research N. Y. (1970).

[8] T. Murata:"Petri nets, marked graphs and circuit-system theory," IEEE CAS Society Newsletter, 11 (June 1971).

[9] T. Murata:"Circuit theoretic analysis and synthesis of marked graphs," IEEE Trans. on CAS, 24, 7, 400-405 (July 1977).

[10]B. O. Nash:"Reachability problems in vector addition systems," Amr. Math. Monthly, 80, 3, 292-295 (May 1973).

ON THE NODE-NUMBERING PROBLEM

STUART R. SOLOWAY
The Codex Corporation, Mansfield, Massachusetts

ABSTRACT

The problem of numbering identical nodes having limited knowledge of the network topology is discussed. Necessary and sufficient conditions are presented for the existence of a non-deterministic node-numbering algorithm, for the existence of a deterministic node-numbering algorithm, and for the existence of a node-numbering algorithm which succeeds with probability 1. A previously open conjecture about graph coverings is also proven.

DEFINITION OF THE PROBLEM

Consider a connected network consisting of a number of identical nodes and bidirectional links joining pairs of not necessarily distinct nodes. Any finite length message may be sent by any node to any adjacent node. We assume messages are always delivered accurately over the links after a transmission delay described by an everywhere finite probability density function. Each node initially knows its own configuration, that is how many links are adjacent to it, and may choose to number these links in any random manner. Such a numbering will be called a ′port numbering′ if it is sequential. Each node also knows certain information about the global network topology. To make this precise, we assume each node knows that the underlying graph (where a graph is allowed to have multiple edges and loops) of the network is isomorphic to some member of a set of graphs C. All nodes have the computing power of a state machine (not necessarily finite) and possesses a true random number generator.

A network is said to have a proper node-numbering if all nodes have entered a halt state and each node has assigned itself a unique integer from 1 to the number of nodes in the network.

The significance of the node-numbering problem is that it seems to be the toughest topological problem to solve with a distributed algorithm running on a network of identical nodes. If each node in a connected network has a unique node number, then the network topology can be determined by each node and a unique ′leader′ can be selected to manage the network. (In fact, the problem of finding a leader is equivalent to that of numbering the nodes in a connected network.)

COVERINGS

The concept of a graph covering will be needed for the following.

We will say a mapping p from connected graph G1 to graph G2 preserves a port numbering of the vertices if for any vertex v1 and incident edge e1 of G1 e1 is incident to v1 via port n1 implies edge p(e1) is incident to vertex p(v1) via port n1. (From now on, all graphs will be assumed to be connected.)

Let p be a mapping of graph G1 onto graph G2. G1 is said to cover G2 via mapping p if p preserves some port numbering of G1 and G2 and if for any vertex v1 in G1 p maps the set of edges incident with v1 onto the set of edges incident with p(v1).

This definition is an extension of that given by (1) to include graphs with multiple edges and loops. A covering is illustrated in Figure 1.

The 'universal covering' of a graph G is defined as the maximal (usually infinite) covering of G. Intuitively, the universal covering of G is the map one would draw of G by following paths from vertex to vertex with no way of determining whether the endpoints of two different paths were the same or not. A universal covering is illustrated in Figure 2.

For vertex v in a graph G we will use the term 'h-neighborhood of v' to mean the set composed of all vertices reachable from v by paths of length h or less and all edges incident with those vertices.

Let G1 and G2 be two graphs with vertex w1 in G1. Let p be a mapping from the h-neighborhood of w1 onto the h-neighborhoood of w2=p(w1). We will say G1 is an h-local covering of G2 via mapping p with centers w1 and w2 if there is some port numbering which is preserved by p and if for any vertex v1 in the h-neighborhood of w1 p maps the set of edges incident with v1 onto the set of edges incident with p(v1). It should be noted that where an edge in the h-neighborhood of G1 is incident to a vertex not in the h-neighborhood of G1 there is no requirement that any port numbering at that vertex be preserved. A 2-local covering is illustrated in Figure 3.

THE BASIC THEOREM

The following theorem imposes a limit on what identical nodes in any network can learn about the network through a distributed algorithm. The limit has nothing to do with the computing power of the nodes themselves, since our computational model is more powerful than a Turing machine, but is due to the fact that the network links have finite bandwidth.

The Basic Theorem - Let G1 be an n-local covering of G2 with centers x1 and x2 in G1 and G2, respectively. Consider two networks with topologies G1 and G2 and consider nodes at vertices x1 and x2. Let algorithm A have non-zero probability of x2 entering some state s in network G2 before n+1 messages have been received in the network. Then algorithm A has non-zero probability of x1 in G1 entering state s.

Proof - Assume the theorem is false. Then for some G1, G2, n, x1, and x2 meeting the requirements of the theorem there is an algorithm A with a state s reachable with non-zero probability at x2 before n+1 messages have been sent but not reachable at x1.

If the above holds, then it also holds if we make the following changes in the way links operate:

Change 1: Transmission delays are still described by a probability density function, but the function is 0 for transmission times less than some value e.

Change 2: Message transmission is synchronous. There is a cosmic clock which ticks every e′ time units where e′<e. A link accepts a message at the clock tick after it is presented to it by an adjacent node and delivers it instantaneously.

Assuming the synchronous links, we prove by induction that s has non-zero probability at x1. We can assume n>0, since the theorem is clearly true for n=0. The theorem also clearly holds until the first clock tick, since x2 has until then received no messages. Assume it holds until tick h-1 and assume message M has non-zero probability of being received by x2 from some adjacent (but not necessarily distinct) y2 on tick h-1 before n+1 messages have been received in the network. Let S be the set of states in which y2 can send M at clock tick h-1 to x2. At tick h-1, S has non-zero probability. Let y1 be adjacent to x1 and mapped by p to y2. Since y1 and y2 are adjacent to x1 and x2, G1 is an (n-1)-local covering of G2 via p with centers y1 and y2. Therefore, by the induction hypothesis, S has non-zero probability at tick h-1 at y1. Also, any set of states S′ having non-zero probability at x2 when M is received has non-zero probability at x1 when M is received from y1. As a result, the conditions of the theorem still hold after M is received and therefore until tick h. Since the conditions of the theorem also hold if no message is received on that clock tick, the theorem is proven.

QED

The reason we have required that link delays be described by a finite probability density function is to eliminate a surreptitious channel by which an infinite amount of information, namely the exact time a message was sent, could be transmitted over a link. If, for example, unrestricted instantaneous transmission were allowed with non-zero probability, then the Basic Theorem would not hold.

A THEOREM ABOUT GRAPH AUTOMORPHISMS

The following theorem will be needed to prove the results for deterministic node-numbering.

Theorem 1 - Let G be a graph with a port numbering. If an automorphism preserves the port numbering, then either all vertices are fixed (mapped to themselves) or none are.

Proof - Let F be an automorphism on G such that not all vertices are fixed. If there is a fixed vertex under F, let v be a fixed vertex adjacent to a non-fixed vertex v′. If port n of v leads to v′, then clearly port n of p(v)=v does not lead to p(v′). So either there is no such v or F does not preserve the port numbering.

QED

DETERMINISTIC NODE-NUMBERING

The term ′deterministic′ as applied to an algorithm here will mean that it always terminates with the desired result before n messages have been sent in the network, where n depends on the network topology and on the algorithm. A deterministic algorithm therefore cannot involve any contention protocol.

If the nodes of a network can be numbered, each node can deterministically find out the topology of the network, but it will be seen that the converse is not true. The necessary and sufficient conditions for a deterministic node-numbering algorithm to exist for some set C of allowable topologies will be derived by first determining the conditions for network topology to be decideable and then determining the conditions for node-numbering if network topology is known.

Theorem 2 - There exists a deterministic distributed algorithm to construct a map of the network topology at each network node if and only if, for each graph G in C, there is a finite integer h such that no graph in C has a universal covering that is an h-local covering of G. By a 'map of the network topology' is meant a graph isomorphic to the underlying graph of the network. Figure 4 shows a set C for which the topology cannot be deterministically decided.

Proof - First we prove that if for some G in C and for any h there exists a G1 whose universal covering is an h-local covering of G then at least one node of the network cannot deterministically decide what the network topology is. Assume G in C and vertex v in G such that for some G1 in C the universal covering U1 of G1 through mapping p1 is an h-local covering of G through mapping p and with centers v and v1. Define an equivalence relation $\underline{=}$ among vertices and edges in the h-neighborhoods of v and v1 as follows: If there exists a vertex or edge u in U1 such that p(u)=x and p1(u)=x1 then $x \underline{=} x1$. If $x \underline{=} y$ and $w \underline{=} x$ then $w \underline{=} y$. Create a graph G* whose vertices and edges are equivalence classes under the '$\underline{=}$' relation. An edge in G* is incident with a vertex if and only if some edge in the equivalence class containing the edge is incident with some vertex in the equivalence class containing the vertex. Let v* be the equivalence class to which v and v1 belong. Clearly G and G1 are both h-local coverings of G*. The centers of the h-local covering by G are v and v* and the centers of the h-local covering by G1 are v1 and v*. Consider networks with topology G, G1, and G* whose nodes all run the same distributed algorithm. By the Basic Theorem any state s reachable at v* with non-zero probability before h+1 messages have been received is reachable with non-zero probability at v and at v1. But h may be arbitrarily large. So no matter how many messages have been received, there is some state that v can be in such that the network topology cannot be distinguished from some G1. This means that node v may never be sure the network topology is indeed G, and the first part of the theorem is proven.

Next we must prove the converse, that if for any G in C an h exists then there is a deterministic algorithm to determine network topology. It is easy to construct a deterministic algorithm by which each node investigates the universal covering farther and farther out from itself. This may be done by having each node choose a port numbering and broadcast messages periodically containing a history of all messages received over each of its links. Eventually each node will find that for some G in C the information it has is consistent with the universal covering of G and all possible paths of the required length h or less are known. Then each node knows the network topology is G.

QED

Assuming that a deterministic algorithm exists to determine network topology, the nodes are numberable if and only if each graph G in C has no graph automorphism without a fixed vertex. We make this a theorem:

Theorem 3 - In a network in which all nodes know the topology, the nodes are deterministically numberable if and only if every graph automorphism on the network topology has a fixed vertex.

Proof - We first show that if there is no automorphism on G without a fixed vertex, then there exists a deterministic node-numbering algorithm for G. The simplest (but not very efficient) way to number the nodes is this: First each node chooses a port numbering. It is easy to implement a protocol by which a node can send a message over any desired path, specified using the port numberings, to any desired remote node. Using such a protocol, each node constructs a graph of the network topology with the port numberings chosen by each node. Then the node specially colors the vertex corresponding to itself in the graph. By Theorem 1 no non-trivial automorphism on the graph preserves the port numbering, so this may be done unambiguously. There exists a function which, given a graph from C with port numberings, returns a proper node numbering for the graph. Such a function is now used by each node to find its unique number.

Next we show the converse, that if there is a non-identity automorphism F on graph G without a fixed vertex, then there is no deterministic node-numbering algorithm for a network with topology G. By logic similar to that used in proving the Basic Theorem, it can be shown that if the statement is true in a network with any link delay probability distribution, then it is true in a network with synchronous links as described in that proof. Therefore, assume that the network has synchronous links. It is clearly possible that at each clock tick and for all adjacent pairs of vertices v and v′, if a message travels from vertex v over edge e to vertex v′, then the same message travels from vertex F(v) over edge F(e) to vertex F(v′). If this happens, it is impossible for a vertex v ever to be certain it has chosen a different node number from the vertex F(v).

QED

Theorems 2 and 3 together give necessary and sufficient condition on C for there to exist a deterministic node-numbering algorithm for network topologies in C.

It is interesting to note that the above result depends on the choice of model. With the alternative model of (1) it is not valid because that model allows two adjacent nodes to arbitrate between each other with a sort of ′coin toss′. However, the result about recognizing graph topology is independent of the model.

NON-DETERMINISTIC NODE-NUMBERING

The term ′non-deterministic′ as applied to an algorithm here will mean that the algorithm can (regardless of how low the probability) complete with the desired result and cannot complete with an incorrect result. It will be seen that a non-deterministic node-numbering algorithm exists for a set C of allowable network topologies if and only if an algorithm exists which succeeds with probability 1.

We now show necessary and sufficient conditions for there to be a non-deterministic algorithm to number nodes, given that the network topology is in some set C.

Theorem 4 - There is a non-deterministic algorithm to number nodes for networks with topology in set C if and only if for every graph G in C there exists an h such that no other graph in C is an h-local covering of G. Further, such an algorithm exists if and only if a non-deterministic algorithm exists which terminates with probability 1.

Proof - First we show that if such h′s exist, then a non-deterministic node-numbering algorithm which succeeds with probability 1 exists. To do this we will describe such an algorithm. In the following, h(G) will mean the h associated with graph G in C. First we construct an algorithm called A(n) for n>0 which will be used as a building block in constructing the needed algorithm:

> In A(n) nodes have names of length n which they assign themselves randomly at initialization. The names are constructed from some arbitrary finite alphabet. Messages consisting of lists of ′records′ are exchanged among adjacent nodes. A ′record′ has this format: (x, a1, a2, ...) where x is the name of the node originating the record and ai is the name of the node adjacent by port number i. We will say a record is about node x to mean it is created by a node with name ′x′.
>
> All nodes initially choose names and exchange them with all adjacent nodes. Then each node broadcasts a message with one record about itself to each neighbor other than the neighbor from which the message was received. Each node on receiving a message containing less than n records appends a record about itself and rebroadcasts the message to all neighbors. Thus the messages are traces of all possible walks of length less than n through the network such that no link is used twice in succession.
>
> By the ′length′ of a message we will mean the number of records in it. A(n) completes at a node when the node has received and rebroadcast as needed all possible messages of length 1 through n from all adjacent nodes. It knows it has received all possible messages of length k when it has received length 1 messages from all neighbors, it has received all possible messages of length k-1, and for every length k-1 message it has received a length k message prefixed by it for each adjacent node listed in the last record. Since a node never completes A(n) until after it has sent out any messages a neighbor might be waiting for to complete A(n), all nodes complete A(n).
>
> Each node then tries to construct a graph and decide whether or not it knows the network topology. Before constructing the graph it checks all the records it has received for ′consistency′. For all node names a and b if there is a record about node a in which b is listed as an adjacent node k times then there must be a record about node b in which a is listed k times as an adjacent node. Also, all records about node a must be identical. If either test fails the records are considered to be inconsistent and no graph can be built. If all the received records are consistent in this sense, then a graph can be built in which the vertices are labelled

with the node names and vertices are connected as described in the received messages.

If a graph is built, if it is isomorphic to some G in C, and if n is greater than h(G), then G is isomorphic to the underlying graph of the network. This is so because by the way the graph is built the function p which takes a node to the vertex with its label is a h-local covering of the graph G by the true network topology for h=h(G). So, by the restrictions we have placed on C, G is isomorphic to the true network topology.

Further, if a node decides that it knows the network topology, then all other nodes make the same decision and each node has a unique name for itself as well as for all other nodes in the network. It is therefore a simple matter to construct a proper node numbering by any arbitrary ordering of node names.

Now we construct an algorithm as follows: Each node first starts algorithm A′(1) where A′(n) is A(n) modified by inserting the number n before each message. When executing algorithm A′(n) messages prefixed with n+1 instead of n are saved to be processed when and if A′(n+1) is started. After completion of algorithm A′(n), if a proper node numbering is not known at a node (and therefore at all nodes) algorithm A′(n+1) is started. Since it is clear that each node will eventually be able to construct a graph of the correct network topology and that no node will ever construct any other graph, this algorithm does what is required.

Now we prove the converse. In fact, we prove something stronger: If there exists a non-deterministic node-numbering algorithm then the h-local covering condition holds. The proof is by contradiction. Let G be in C and assume that for any v in G and any h there is a G1 in C which is an h-local covering of G with center v in G. By the Basic Theorem, until h messages have been received in the network any state assumed by v might also be assumed by some v1 in a network with some topology G1. Therefore, there is no state which can be assumed by node v if and only if the network topology is indeed G, and no non-deterministic node-numbering algorithm exists. This completes the proof of Theorem 4.

QED

RESTRICTIONS ON NETWORK SIZE

Let us assume that C, the set of allowable network topologies, contains all possible connected graphs having numbers of vertices in some set N. In other words, let the algorithm know beforehand that the number of nodes in the connected network is restricted in some way but let there be no other topological restrictions. For nontrivial N there is no deterministic node-numbering algorithm, since for any integer n greater than 1 a ring network of n nodes cannot be deterministically numbered due to Theorem 3. As a special case of Theorem 4, there is a non-deterministic node-numbering algorithm for connected networks of restricted size if and only if the following conditions of N are met:

1 - N is finite. If N were infinite, an infinite number of ring networks would be in C, and that alone would make node-numbering impossible.

2 - No member of N is a divisor of any other member of N. If this were not so, one member of C could cover another.

If the above two assertions holds, no member of C can cover any other member. Further, a member of C can be a h-local covering of another member for h greater than the maximum member of N only if it is a covering of that member. So it can be seen that the above conditions are both necessary and sufficient.

For the case in which N contains one element, meaning that the algorithm knows in advance the number of nodes, a node-numbering algorithm is described in (2).

DISTINGUISHING NETWORK TOPOLOGIES AND A THEOREM ABOUT COVERINGS

A problem closely related to node-numbering is that of distinguishing between pairs of network topologies with a distributed algorithm.

(1) poses the following conjecture about graph coverings:

> Conjecture 1 - If G and J in G′ (the set of all finite, loopless, non-trivial connected graphs without multiple edges) have isomorphic universal coverings, then they have a finite common covering.

It is then shown that, assuming the conjecture holds, it is possible to distinguish between two possible network topologies in G′ with a deterministic algorithm if and only if the two topologies have non-isomorphic universal coverings. We now prove the conjecture:

Theorem 5 - The above conjecture holds and may be generalized to allow G′ to include graphs with multiple edges and loops.

Proof: Let U be a universal covering of G via mapping p1 and J via mapping p2. The ′ID′ of each node in U will be (x,y) where x is its image in G under p1 and y is its image in J under p2. We will gradually modify U by ′marking′ vertices, rerouting edges, and drawing new edges until it contains a component which is a finite common covering of G and J.

The first step in this process is to choose a random vertex of U and mark it.

Next we repeat an iterative procedure, described below, until the algorithm is done. At the start of each iteration the following statements are true:

1 - The marked vertices are all in the same component of the modified U.

2 - The mappings p1′ and p2′, which are p1 and p2 restricted to that component, are coverings of G and J.

3 - No two marked vertices have the same ID.

4 - No unmarked vertex has more than one link to any one other vertex.

It will be seen these statements are true both before and after each iteration and that therefore the algorithm is correct. The iterated procedure is as follows:

> If there is no unmarked vertex adjacent to any marked vertex, then there is one finite component in which all the marked vertices lie, and it is a finite common cover of G and J. The algorithm is then complete.
>
> Otherwise choose some unmarked vertex v adjacent to a marked vertex v′. If there is no marked vertex with the same ID, mark it and go on to the next iteration, and note that the statments made at the start of the iteration still hold. Otherwise, there is a marked vertex v′′ with the same ID as v. Since p1′ and p2′ are coverings, v′′ has an edge to some unmarked v′′′ with the same ID as v′. (It must be unmarked because two unmarked vertices cannot have the same ID.) Erase the edge linking v′′ and v′′′. Disconnect from v the edge linking v and v′ and reconnect the loose end to v′′. Any port numbering preserved by p1′ or p2′ before the change is still preserved, so p1′ and p2′ are still coverings. Continue with the next iteration, again noting that the statements made at the start of the iteration still hold.

QED

It may also be shown that two network topologies may be distinguished by a non-deterministic algorithm if and only if neither is a covering of the other. The results given here for distinguishing network topologies may also be derived directly from Theorems 3 and 5.

ACKNOWLEDGEMENT

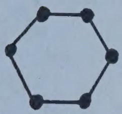
The author wishes to thank Professor Pierre Humblet of MIT and Codex for his help.

REFERENCES

[2] Dana Angluin, "Local and Global Properties in Networks of Processors." Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing. ACM Order Number 508800. 1980.

[2] Maurice F. Aburdene, "Numbering the Nodes of a Graph by Distributed Algorithms." Proceedings of the Seventeenth Annual Allerton Conference on Communication, Control, and Computing. 1979.

Figure 1



covers

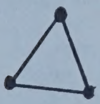
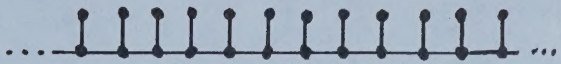


Figure 2



Graph above is (infinite) universal covering of graph below:

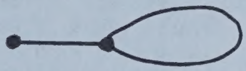
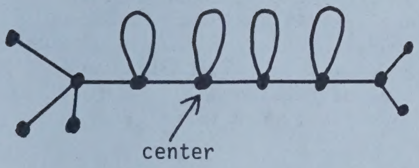


Figure 3



Graph at left is a 2-local covering of graph at right

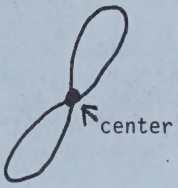
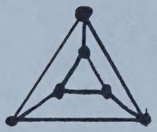
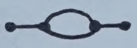


Figure 4

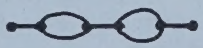
If C contains graph A and graphs B1, B2, ...Bn for all positive integer n then topology cannot be deterministically



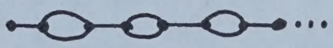
A



B1



B2



B3

decided because the universal covering of Bn for large n looks locally like a 3-regular tree, which covers A.

A BINOMIAL MODEL FOR SOFTWARE PERFORMANCE PREDICTION

J. G. SHANTHIKUMAR
Dept. of Industrial Engineering
and Operations Research
Syracuse University
Syracuse, New York

ABSTRACT

The main intent of this paper is to derive a Binomial model for software performance prediction for a software system with several types of errors. Starting from some basic model assumptions and using a simple approach we develop a Binomial model for software performance prediction. Using some results derived in a previous paper we give expressions for performance prediction. In a supplementary report, we establish an equivalence between a special case of this Binomial model and the Imperfect Debugging Model of Goel and Okumoto, and give some numerical results to illustrate performance measures.

1. INTRODUCTION

During the past few years, considerable interest has been shown in studying software failure phenomenon, with the objective of developing models for software performance prediction. These studies fall primarily into two categories. They are (i) empirical analyses of collected data, and (ii) analytical models for quantitative assessment of software performance. In a recent article [6], Schick and Wolverton have discussed and compared most of these models. They have also given an extensive bibliography of software reliability research.

The purpose of this paper is to derive expressions for software performance prediction for a software system with several types of errors using a Binomial model. In a supplementary report [8], we exhibit the equivalence between a special case of this Binomial model and the Imperfect Debugging model of Goel and Okumoto [3]. The description of the model assumptions and the system performance measures of interest are given in section 2. A Binomial model based on the model assumptions is derived in section 3. In sections 4 and 5, expressions for performance prediction and for the estimation of model parameters for one type of error are derived using this Binomial model. Overall system performance measures accounting for all types of errors are presented in section 6. Section 7 contains expressions for performance prediction.

2. THE MODEL

The model class of software systems considered here is assumed to satisfy the following:

(1) There are m types of errors that cause software failures.
(2) The initial error content N_j (at the beginning of the observation phase, that is at time t=0) of type j errors of this software system is a fixed unknown constant, for all j=1,2,...,m.

(3) The probability that a type j error will cause a software failure in a small time interval $(t,t+\Delta t)$ is equal to $\lambda_j \Delta t + O(\Delta t)$, where $\lim_{\Delta t \to 0} O(\Delta t)/\Delta t = 0$. These probabilities for all errors are independent of one another and independent of time t.

(4) When a type j error occurs, it is detected and corrected with probability p_j. That is, with probability q_j $(q_j=1-p_j)$, the error is imperfectly debugged (that is, the error is not eliminated).

(5) No new errors are created, at most one error is removed at a correction time, and the time taken to detect and correct an error is negligible.

With this set of assumptions we will now formulate our model. The purpose of this model is to derive the system performance measures assuming that no observations have been made. These performance measures are such that the model parameters N_j, p_j, and λ_j, $j=1,2,\ldots,m$, can be estimated from them and the observed data up to some time, say t_ℓ. So we will next describe the performance measures that are of interest to us.

2.1 The System Performance Measures

The performance measures that are of interest to us are divided into two primary categories. The system performance measures to be used in performance prediction form one, and the performance measures to be used in parameter estimation form the other category.

2.1.1 Performance Measures for Prediction

(i) Distribution of number of remaining errors at time t
(ii) Distribution of time to a specified number of remaining errors
(iii) Distribution of time to a completely debugged system
(iv) Distribution of time to the first software failure

2.1.2 Performance Measures to be Used for Parameter Estimation

(i) Expected number of errors detected by time t
(ii) Expected number of distinct errors detected by time t

It should be noted that Goel and Okumoto [2, 3] and Shanthikumar [7] used expected number of imperfect debuggings to derive an estimator for p_j. In their derivation, however, the expected number of imperfect debuggings is approximated by the expected number of imperfect debuggings detected.

3. THE BINOMIAL MODEL FOR TYPE j ERRORS

In this section we will derive a Binomial model for the type j error of the software system based on the assumptions given in section 2. We shall do this by considering each error of each type separately. Let us consider a type j error (out of the total N_j type j errors) present in a software at time t=0. Assumption (2) guarantees that the time to a software failure caused by this error is exponentially distributed with mean $1/\lambda_j$. We will first find the probability that this error will be perfectly debugged before time t.

Assume that this error occurred R_j times before it is perfectly debugged. Then

$$P(R_j = n) = p_j q_j^{n-1} \quad , \quad n=1,2,\ldots, \tag{1}$$

Now let T_j be the time of perfect debugging of this error. Then T_j is the random sum of R_j exponential random variables with mean $1/\lambda_j$. So conditioning on R_j we have an Erlang-n distribution for T_j. That is,

$$P(T_j \leq t | R_j = n) = \int_o^t \frac{e^{-\lambda_j x} (\lambda_j x)^{n-1}}{(n-1)!} \lambda_j dx \tag{2}$$

Now unconditioning with respect to R_j, from (1) and (2) we get

$$P(T_j \leq t) = \sum_{n=1}^{\infty} p_j q_j^{n-1} \int_o^t \frac{e^{-\lambda_j x} (\lambda_j x)^{n-1}}{(n-1)!} \lambda_j dx, \quad t \geq 0 \tag{3}$$

$$= 1 - e^{-\lambda_j p_j t} \quad , \quad t \geq 0 \tag{4}$$

is obtained by interchanging the summation and integration and then simplifying equation (3). Then if we define $a_j(t) = \exp(-\lambda_j p_j t)$, we have

$$P(\text{the type j error is perfectly debugged by time t}) = P(T_j \leq t)$$
$$= 1 - a_j(t) \quad , \quad t \geq 0 \tag{5}$$

$$P(\text{the type j error is not perfectly debugged by time t})$$
$$= P(T_j > t) = a_j(t) \quad , \quad t \geq 0 \tag{6}$$

and $P(\text{the type j error appears at least once by time t})$
$$= 1 - \exp(-\lambda_j t), \quad t \geq 0 \tag{7}$$

With these three results (5), (6), and (7) we have Binomial distributions for the number of type j errors remaining at time t, for the number of type j errors perfectly debugged by time t, and for the number of distinct type j errors detected by time t with parameter sets $(N_j, a_j(t))$, $(N_j, 1-a_j(t))$ and $(N_j, 1-\exp(-\lambda_j t))$, respectively.

4. PERFORMANCE MEASURES FOR TYPE j ERRORS

In this section, using results (5), (6), and (7) we will derive performance measures that are of interest to us.

4.1 Distribution of Number of Remaining Errors

Let $P_{N_j,n_j}(t)$ be the probability that there are n_j type j errors remaining at time t. Since the probability that a type j error is not perfectly debugged by time t is $a_j(t)$ (see equation (6)), the probability that n_j out of N_j errors initially present at time t=0 is not perfectly debugged by time t is (Binomial) given by,

$$P_{N_j,n_j}(t) = \binom{N_j}{n_j}(a_j(t))^{n_j}(1-a_j(t))^{N_j-n_j}, \; n_j=1,2,\ldots,N_j \qquad (8)$$

with mean

$$E(N_j(t)) = N_j \exp(-\lambda_j p_j t) \qquad (9)$$

A software model satisfying conditions given in section 2 with one type of error should be identical to the imperfect debugging model of Goel and Okumoto [3]. Even though the results for $P_{N,n_o}(t)$ given in [3] seems different from equation (8), in [8] we have established their equivalences.

4.2 Distribution of Time To A Specified Number of Remaining Errors

Let $G_{N_j,n_j}(t)$ be the probability that the time required to n_j number of type j remaining errors is less than or equal to t. If the time required to n_j number of type j remaining errors is less than or equal to t, the number of type j errors remaining at time t should be less than or equal to n_j. Then from standard probability arguments we have

$$G_{N_j,n_j}(t) = \sum_{n=0}^{n_j} P_{N_j,n}(t) = \sum_{n=0}^{n_j} \binom{N_j}{n}(a_j(t))^{n}(1-a_j(t))^{N_j-n}, \qquad (10)$$

$$n_j=1,2,\ldots,N_j.$$

4.3 Distribution of Time to a System Completely Debugged of Type j Errors

The system is completely debugged of type j errors if all N_j type j errors are removed. Then the probability $G_{N_j,0}(t)$, that the system is completely debugged of type j errors by time t is given by,

$$G_{N_j,0}(t) = (1 - a_j(t))^{N_j}, \quad t \geq 0 \qquad (11)$$

since $(1-a_j(t))$ is the probability that a type j error is perfectly debugged by time t (see equation (5)).

4.4 Distribution of Time to the First Software Failure Due to a Type j Error

Let $G_j(t)$ be the probability that the time to the first software failure due to a type j error is less than or equal to t. Since the time to the first software failure due to a type j error is the minimum of N_j exponential random variables with mean $1/\lambda_j$, we have,

$$G_j(t) = 1 - \exp(-N_j\lambda_j t), \quad t \geq 0 \tag{12}$$

4.5 Expected Number of Type j Errors Detected by Time t

Let us consider one out of the N_j type j errors present at time t=0. Let $N_j'(t)$ and $N_j''(t)$ be the number of times this error showed up before and after time t before its perfect debugging. Note that $N_j'(t)$ and $N_j''(t)$ are random variables and $N_j''(t)$ will take a value of zero if the error is perfectly debugged before time t. Then the total number of software failures caused by this error is R_j (as defined by equation (1)) equal to $N_j'(t) + N_j''(t)$. Since this total has a geometric distribution with parameter p_j, we have,

$$E(R_j) = E(N_j'(t)) + E(N_j''(t)) = 1/p_j \tag{13}$$

Since the geometric distribution is memoryless, given that the perfect debugging occurred after time t, the expected number of software failures occurred after time t due to this error before its perfect debugging is also $1/p_j$. However, the probability that this error is perfectly debugged after time t is $a_j(t)$, (see equation (6)). Then from (13) and $E(N_j'') = a_j(t)/p_j$ we get

$$E(N_j'(t)) = (1 - a_j(t))/p_j \tag{14}$$

Therefore the total expected number $M_{N_j}(t)$ of type j errors detected by time t is, from (13) and (14), given by

$$M_{N_j}(t) = N_j(1 - a_j(t))/p_j, \quad t \geq 0 \tag{15}$$

since

$$M_{N_j}(t) = N_j E(N_j'(t)).$$

4.6 Expected Number of Distinct Type j Errors Detected by Time t

Let $N_{D_j}(t)$ be the expected number of distinct type j errors detected by time t. In section 3 (see equation (7)) we have given that the probability of a type j error appearing at least once before time t is $(1 - \exp(-\lambda_j t))$. Then it is easily seen that

$$N_{D_j}(t) = N_j(1 - \exp(-\lambda_j t)) , \quad t \geq 0 \tag{16}$$

5. ESTIMATION OF MODEL PARAMETERS

Several estimation procedures may be developed to obtain the estimates $\hat{N}_j$, $\hat{p}_j$, and $\hat{\lambda}_j$ of N_j, p_j, and λ_j, respectively, using the expression derived for performance measures. In this paper, however, we will restrict ourselves to one such method, proposed here (Comparison of this method with that proposed by Goel and Okumoto [3] and others, using a simulation model will be reported elsewhere).

It should be noted that this method is proposed assuming that the data available is in sets of triplets $(u_j(i), v_j(i), t(i))$, $i=1,2,\ldots,n$; $j=1,2,\ldots,m$, where $u_j(i)$ and $v_j(i)$ are, respectively, the cumulative number of software failures and imperfect debuggings detected due to type j errors during the time interval $(0,t(i)]$. Then the cumulative number of distinct type j errors $w_j(i)$ detected by time t(i) is

$$w_j(i) = u_j(i) - v_j(i), \quad i=1,2,\ldots,n; \; j=1,2,\ldots,m . \tag{17}$$

5.1 The Method

Using equations (15) and (16) it can be shown that some estimates for N_j, λ_j and p_j are

$$\hat{N}_j, \; \hat{\lambda}_j, \text{ and } \hat{p}_j = \{\frac{\hat{N}_j}{\hat{a}_j}\} \tag{18}$$

where $\hat{N}_j$ and $\hat{\lambda}_j$ are solutions of

$$\frac{\partial R_j}{\partial N_j} = 0 , \quad \frac{\partial R_j}{\partial \lambda_j} = 0$$

and

$$R_j = \sum_{i=1}^{n} (w_j(i)-w_j(i-1)-N_j(\exp(-\lambda_j t(i-1))-\exp(-\lambda_j t(i))))^2$$

and $\hat{a}_j$ and $\hat{b}_j$ are solutions of

$$\frac{\partial S_j}{\partial a_j} = 0 , \quad \frac{\partial S_j}{\partial b_j} = 0$$

and

$$S_j = \sum_{i=1}^{n} (u_j(i)-u_j(i-1)-a_j(\exp(-b_jt(i-1))-\exp(-b_jt(i))))^2$$

6. OVERALL PERFORMANCE MEASURES

Using the results derived in section 4, we will present the results for overall performance measures.

6.1 Distribution of Number of Remaining Errors

Let $P_{\underset{\sim}{N},\underset{\sim}{n}}(t)$ be the probability that there are $\underset{\sim}{n} = (n_1,n_2,\ldots,n_m)$ number of errors (n_j of type j errors) remaining at time t. Then from (8),

$$P_{\underset{\sim}{N},\underset{\sim}{n}}(t) = \prod_{j=1}^{m} \binom{N_j}{n_j}(a_j(t))^{n_j}(1-a_j(t))^{N_j-n_j} , \tag{19}$$

and mean $E(N_j(t)) = N_ja_j(t)$ of type j errors remaining, where $\underset{\sim}{N} = (N_1,N_2,\ldots,N_m)$ and $a_j(t) = \exp(-\lambda_jp_jt)$, $j=1,2,\ldots,m$.

6.2 Distribution of Time to a Specified Number of Remaining Errors ($G_{\underset{\sim}{N},\underset{\sim}{n}}(t)$)

From (10) we can show that

$$G_{\underset{\sim}{N},\underset{\sim}{n}}(t) = \prod_{j=1}^{m} \left(\sum_{n=0}^{n_j} \binom{N_j}{n}(a_j(t))^{n}(1-a_j(t))^{N_j-n} \right) \tag{20}$$

$$n_j=1,2,\ldots,N_j ; \; j=1,2,\ldots,m.$$

6.3 Distribution of Time to a Completely Debugged System

Let $G_{\underset{\sim}{N},\underset{\sim}{0}}(t)$ be the probability that the system is completely debugged by time t. Then from (11),

$$G_{\underset{\sim}{N},\underset{\sim}{0}}(t) = \prod_{j=1}^{m} (1 - a_j(t))^{N_j} , \; t \geq 0 \tag{21}$$

6.4 Distribution G(t) of Time to the First Software Failure

From $G_j(t)$ as defined in section 4 we have

$$\begin{aligned} G(t) &= 1 - \prod_{j=1}^{m} \exp(-N_j\lambda_jt), \quad t \geq 0 \\ &= 1 - \exp(-\widetilde{\widetilde{N\lambda}}t), \quad t \geq 0 \end{aligned} \tag{22}$$

where

$$\underset{\sim}{N}\underset{\sim}{\lambda} = \sum_{j=1}^{m} N_j \lambda_j$$

7. PERFORMANCE PREDICTION

Using the expressions derived for overall performance measures in section 6, the observed data and the estimated parameters (see section 5), expressions for performance prediction can be easily developed. In this section we will present these results (by adopting the results given in Shanthikumar [7]).

7.1 Distribution of the Number of Remaining Errors After $\underset{\sim}{\ell}$ Software Failures

Suppose we have observed $\underset{\sim}{\ell} = (\ell_j,\ j=1,2,\dots,m)$, $\ell_j = u_j(n)$, $j=1,2,\dots,m$, software failures up to time $t_\ell = t(n)$ under the given set of data $(u_j(i),\ v_j(i),\ t(i))$, $i=1,2,\dots,n$; $j=1,2,\dots,m$. Then the number k_j of distinct type j errors detected up to time t_ℓ is equal to $w_j(n) = u_j(n) - v_j(n)$. Then the probability $P_{\underset{\sim}{N},\underset{\sim}{\tau}}(t_\ell)$ that the number of remaining errors at time t_ℓ is $\underset{\sim}{\tau} = (\tau_j,\ j=1,2,\dots,m)$, τ_j of type j, is approximated by a product of Binomial form, (see [7]),

$$P_{\underset{\sim}{N},\underset{\sim}{\tau}}(t_\ell) = \prod_{j=1}^{m} \binom{k_j}{N_j-\tau_j} b_j^{N_j-\tau_j} (1-b_j)^{k_j+\tau_j-N_j}, \tag{23}$$

$$\tau_j = N_j, N_j-1, \dots, N_j-k_j \geq 0;\ j=1,2,\dots,m$$

with mean

$$E(N_j(t_\ell)) = N_j - b_j k_j \tag{24}$$

of type j errors, and

$$E(N(t_\ell)) = \sum_{j=1}^{m} (N_j - b_j k_j) \tag{25}$$

of total errors, where

$$b_j \overset{\sim}{=} \frac{1}{w_j(n)} \sum_{i=1}^{n} (w_j(i) - w_j(i-1)) \left\{ \frac{\hat{p}_j}{\hat{p}_j + \hat{q}_j \exp(-\hat{\lambda}_j \hat{p}_j (t(n)-t(i)))} \right\}$$

$$w_j(0)=0,\ j=1,2,\dots,m.$$

7.2 Distribution of Number of Remaining Errors at Time $t+t_\ell$

From equation (19) and (23) it follows that the probability $P_{\underset{\sim}{N},\underset{\sim}{r}}(t_\ell+t)$ of $\underset{\sim}{r}$ remaining errors at time $t_\ell+t$ is given by

$$P_{\underset{\sim}{N},\underset{\sim}{r}}(t_\ell+t) = \prod_{j=1}^{m} \sum_{i=N_j-k_j}^{N_j} \binom{i}{r_j}(a_j(t))^{r_j}(1-a_j(t))^{i-r_j}$$

$$\binom{k_j}{N_j-i} b_j^{N_j-i}(1-b_j)^{k_j+i-N_j} \quad , \tag{26}$$

$$0 \le r_j \le N_j,\ j=1,2,\dots,m,\ \binom{x}{y}=0,\ x < y,$$

with expected value

$$E(N_j(t_\ell+t)) = (N_j-b_jk_j)\ \exp(-\lambda_j p_j t) \tag{27}$$

to type j errors and

$$E(N(t_\ell+t)) = \sum_{j=1}^{m} (N_j-b_jk_j)\exp(-\lambda_j p_j t) \tag{28}$$

of total errors.

7.3 Distribution of Time to a Specified Number of Remaining Errors

Let $G_{\underset{\sim}{N},\underset{\sim}{r}}(t)$ be the probability that before time $t_\ell+t$, the number of type j remaining errors will be r_j, $j=1,2,\dots,m$. Then from (20) and (23) we can show that

$$G_{\underset{\sim}{N},\underset{\sim}{r}}(t) = \prod_{j=1}^{m} \sum_{d_j=0}^{r_j} \Big(\sum_{i=N_j-k_j}^{N_j} \binom{i}{d_j}(a_j(t))^{d_j}(1-a_j(t))^{i-d_j}$$

$$\binom{k_j}{N_j-i} b_j^{N_j-i}(1-b_j)^{k_j+i-N_j}\Big), \quad t \ge 0 \tag{29}$$

7.4 Distribution of Time to a Completely Debugged System

From equation (29), after some algebra, we get

$$G_{\underset{\sim}{N},\underset{\sim}{0}}(t) = \prod_{j=1}^{m} (1-a_j(t))^{N_j-k_j}(b_j+(1-b_j)(1-a_j(t)))^{k_j},\ t \ge 0 \tag{30}$$

7.5 Distribution of Time to Next Software Failure After $\underset{\sim}{\ell}$ Software Failures

Let $\phi_{\underset{\sim}{k}}(t)$ be the probability that the next software failure after $\underset{\sim}{\ell}$ software failures caused by k_j of distinct type j errors, $j=1,2,\dots,m$, have occurred, will occur during $(t_\ell, t_\ell+t]$. Then it can be shown from equations (22) and (23) that

$$\phi_{\underset{\sim}{k}}(t) = 1 - \prod_{j=1}^{m} \exp(-N_j\lambda_j t)(1-b_j+b_j\exp(\lambda_j t))^{k_j},\ t \ge 0 \tag{31}$$

Then the reliability function $R_{\underset{\sim}{k}}(t) \triangleq 1 - \phi_{\underset{\sim}{k}}(t)$ is given by

$$R_{\underset{\sim}{k}}(t) = \prod_{j=1}^{m} (1-b_j+b_j\exp(\lambda_j t))^{k_j} \exp(-N_j\lambda_j t), \quad t \geq 0 \tag{32}$$

and a suitable approximation is (see [7]),

$$R_{\underset{\sim}{k}}(t) \overset{\sim}{=} \exp(-(\widetilde{N\lambda} - \widetilde{bk\lambda})t), \quad t \geq 0, \tag{33}$$

where $\widetilde{N\lambda} = \sum_{j=1}^{m} N_j\lambda_j$

is as defined earlier, and

$$\widetilde{bk\lambda} = \sum_{j=1}^{m} b_j k_j \lambda_j \; .$$

8. CONCLUSION

In this paper, starting from some basic model assumptions and using a simple approach, we developed a Binomial model for software performance prediction for a software with several types of errors. It should be noted that, if one were to use a Semi-Markov model for a software system with several types of errors, the task of solving the model equations will be horrendous (see Goel and Okumoto [3] for the basic model) and most probably the final expressions will not be computationally attractive.

REFERENCES

[1] Akiyama, F., "An Example of Software Debugging," 1971 IFIP Congress, (1971), pp. 353-359.

[2] Goel, A.L. and K. Okumoto, "An Analysis of Recurrent Software Errors in a Real-Time Control System," Proc. of ACM (1978), pp. 496-501.

[3] Goel, A.L. and K. Okumoto, "A Markovian Model for Reliability and Other Performance Measures of Software Systems," AFIPS Conference (1979), pp. 769-774.

[4] Jelinski, Z. and P. Moranda, "Software Reliability Research," Statistical Computer Performance Evaluation, W. Freiberger (Ed.), Academic Press, (1972), pp. 465-484.

[5] Littlewood, B., "A Reliability Model for Markov Structured Software," Applied Statistics, Vol. 24 (1974), pp. 172-177.

[6] Schick, G.J. and R.W. Wolverton, "An Analysis of Competing Software Reliability Models," IEEE Transactions on Software Engineering (1978), pp. 104-120.

[7] Shanthikumar, J.G., "Software Performance Prediction Using a State-Dependent Error Occurrence-Rate Model," Proc. 19th Annual Technical Symposium - Pathways to System Integrity (1980), pp. 67-73.

[8] Shanthikumar, J.G., "A Binomial Model for Software Performance Prediction," Working Paper #80-014, Dept. of Indus. Engr. & Oper. Res., Syracuse University, Syracuse, NY, 13210 (1980).

PRELIMINARY DESIGN OF A HIGHLY PARALLEL ARCHITECTURE FOR REAL-TIME APPLICATIONS

MARK E. BROWN * AND BRUCE W. WEIDE
Department of Computer and Information Science
The Ohio State University
Columbus, Ohio 43210

ABSTRACT

Failure to acknowledge problems in the areas of man-machine interactions and the inadequacies of single processor machines have yielded many bewildering environments for the programmer of real-time applications. These programmers, who find themselves trying to cope with complicated synchronization and communication primitives, are usually engineers with little background in computer science. We propose the use of real-time data-flow graphs to permit more systematic and simpler development of solutions to real-time acquisition and control applications. This is accomplished by narrowing the semantic gap through the similarity of these graphs to those already used in control theory. The use of data-flow also provides automatic interprocess synchronization and communication. Because the real-time data-flow primitives have simple computational requirements, they can be implemented in hardware in a form suitable for fabrication in VLSI.

1. INTRODUCTION

The problems facing real-time application programmers today fall into the categories of man-machine interaction, software, and hardware. In the man-machine interaction category, the first problem is the lack of background of the programmers. These programmers normally have an engineering background with little training in computer science, perhaps only a single Fortran course. Consequently, they are not always prepared to work with the complicated synchronization and communication primitives used in the real-time environment. As if learning these primitives is not a great enough hardship, current technology only allows one to describe parallel tasks by writing what looks like ordinary sequential code that is supposed to do interrupt handling and time management as well as common calculation.

As for the language problem, the advent of high level languages like Concurrent Pascal [2], Modula [7] and Ada [6] has produced an alternative to programming in assembler or Fortran. It is interesting that most real-time programming today is still done in these older, less flexible languages. However, even if the newer langauges were used, the programmer would still be confronted with synchronization primitives and sequential representations of parallel tasks. For instance, Modula (a language originally written for process control) has no built-in method to deal with time [1].

Current hardware also suffers from many inadequacies. While single processor machines are capable of handling small sets of tasks, an increasing number of time dependent tasks will cause the machine to reach its physical limitations. This results in the loss of data or control due to missing the events that schedule the processes. The traditional parallel processing solutions to this problem have problems,

---

This research supported in part by NSF MCS-79-12688.
* Also with Bell Laboratories, Columbus, Ohio.

too. Parallel processor configurations usually provide one processor as a supervisor and others as slaves, in which case time consuming contention problems can arise in shared memories and buses. The master processor may not be capable of scheduling very time critical processes due to its sequential nature. A solution to the single master and common memory/bus problem seems, at first, to lie in distributed control. This leads to complicated synchronization problems, requiring time-stamped messages between processes with their own clocks [5]. The number of messages required for synchronizing the clocks and the overhead involved with processing these messages may offset the advantages of totally distributed control.

A solution to all of these problems can be derived from an alternate form of computation. Until now all of the solutions were based on control-flow. The solution proposed in this paper is based on data-flow [3]. A program in a data-flow language is a directed graph in which the nodes (hereafter referred to as boxes) are operators, and arcs are paths along which values (conveyed by tokens) may travel. A box may fire (execute its operation) according to some rule regarding receipt of tokens on its input arcs and existence of tokens on its output arcs. In theory, at least, a box may be a general finite-state machine with transitions triggered by arrival of input tokens (which are consumed when a box changes state or completes execution) or by consumption of output tokens by other boxes. In practice, however, boxes do not complete execution until all output arcs are empty, i.e., have no tokens on them. Therefore, all operations are scheduled based upon the availability of data required or produced by the respective boxes. This allows a great deal of parallelism to occur during the execution of a program.

2. A REAL-TIME DATA-FLOW LANGUAGE

To further explain the concept of data-flow languages and how they can be used to solve problems in the real-time domain, a more detailed description is required. This section will give some basic definitions and then elaborate with a description of the real-time data-flow language, a program development system, and example program.

2.1 Definitions

As mentioned before, a data-flow program consists of a directed graph. The arcs of the graph represent the paths on which data will flow in the program. The various types of tagged data are transported along these paths by tokens. The token values found to be useful for the real-time data-flow system are listed in Figure 1, and include several special types, such as "?". Tokens are represented by circles on arcs in a data-flow program.

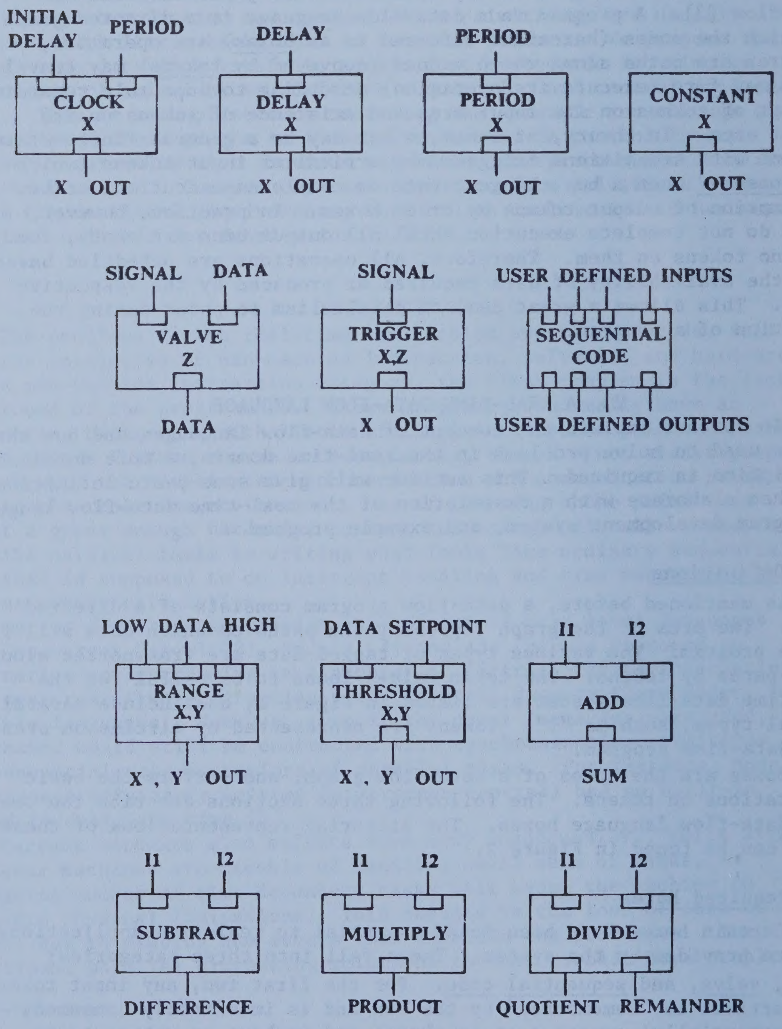
Boxes are the nodes of a data-flow graph, and perform the basic computations on tokens. The following three sections describe the real-time data-flow language boxes. The pictorial representations of these boxes can be found in Figure 2.

2.2 Required Boxes

Certain boxes have been found essential to real-time applications and are provided by the system. These fall into three categories: clock, valve, and sequential code. For the first two, any input token that arrives is "remembered" by the box and is immediately consumed, and (potentially) causes a state change and perhaps an output to be produced.

Representation	Meaning
/	no data token
?	undefined value
+	positive infinity
−	negative infinity
T	boolean true
F	boolean false
numeric	numeric constant
I_n	value of input n

Figure 1 - Token Values and Their Representation in Programs



INITIAL DELAY
PERIOD
CLOCK
X
X OUT
DELAY
DELAY
X
X OUT
PERIOD
PERIOD
X
X OUT
CONSTANT
X
X OUT
SIGNAL
DATA
VALVE
Z
DATA
SIGNAL
TRIGGER
X,Z
X OUT
USER DEFINED INPUTS
SEQUENTIAL CODE
USER DEFINED OUTPUTS
LOW DATA HIGH
RANGE
X,Y
X , Y OUT
DATA SETPOINT
THRESHOLD
X,Y
X , Y OUT
I1
I2
ADD
SUM
I1
I2
SUBTRACT
DIFFERENCE
I1
I2
MULTIPLY
PRODUCT
I1
I2
DIVIDE
QUOTIENT
REMAINDER

FIGURE 2

A clock box has two inputs, an initial "delay", after which it will produce an output token, and a "period" that it will wait before producing additional outputs. The token produced as output by a clock box is described in the interior of the box, specifying either a constant or one of the box's inputs. The clock is used (by a "macro" facility) to produce three other boxes. The delay box is a clock box with its period input always equal to positive infinity. This will cause an output token to be produced after a delay provided on the delay input with no subsequent tokens being produced. The period box is a clock box with its delay input always set equal to the period input token. An output token will be produced by this box at the end of each period. The constant box is a clock box with its period input and delay input always equal to zero. Since the delay and period are zero, the box will produce output tokens as fast as they are consumed by the boxes that follow in the graph.

A valve has a "trigger" token and a "data" token as inputs. Whenever a trigger token is equal to the value specified in the box ("*" matches any value) and there are no tokens on the output arcs, the most recent value received on the data input is output. A trigger box is equivalent to a valve box with a constant specified inside the box indicating the data value being passed on; there is no data input, only the trigger input.

A sequential code box represents an ordinary sequential computation task. It is used to pass information to and from the task and to schedule the task, which is simply sequential code such as a Fortran subprogram. This box will only fire after a token has been received on each of its input arcs and no token exists on any of its output arcs. At the conclusion of the computation, the input tokens are consumed and output tokens produced.

2.3 Additional Useful Boxes

In addition to the required boxes are a set of useful boxes that are provided by the system. These boxes are probably not required for most real-time systems, but they are very helpful.

The range is the first box in this category. It has three inputs and two possible output values listed in the box. The inputs consist of a "high" and "low" bound for the range and a "data" token. On a transition into the bounded area (range) by a data token, the first of the two specified values is output. When there is a transition out of the range the second value is produced. The threshold box is equivalent to a range box with the low token set to negative infinity and the high token referred to as the "setpoint". Other useful boxes do simple arithmetic.

2.4 Supplementary Boxes

The boxes discussed in the previous sections should be sufficient to create real-time data-flow programs to perform most types of common real-time tasks. However, the system is designed to be modular and extensible. Each of the hardware boxes (to be explained further in Section 3.1) will have a writeable control store. When a real-time data-flow program is written, the compiler/constructor will build microprogram modules from a microprogram control library and load them into the hardware. This microprogram control library could be extended by systems programmers to allow for previously undefined boxes. This is not expected to happen frequently and should not be a burden on the every day user of the system, who need not even be aware of this feature.

2.5 The Program Development System

In an effort to improve the man-machine interface, a processor and software will be provided as a program development package. The development system is an ordinary mini- or micro-computer system with a couple of additions: it will require a color graphics display and mouse to draw the data-flow graphs. Colors will be used to signify important factors in the development of the program. For instance, red might represent sequential code boxes that have no code associated with them. Yellow, on the other hand, might characterize such boxes that are partially completed (refined), and green might depict completed sections. White could indicate the box currently being added to the system while it was being moved into position on the screen. The display software will support a few short menus from which the programmer/engineer can choose his/her next item. The system will also have a keyboard for labeling boxes, arcs and tokens. and a hard-copy output device for listing sequential code and for drawing data-flow graphs.

The software will provide additional aids to the programmer. For example, by drawing a closed curve around a section of a graph, a programmer can create a virtual box and store it in a library for use in other tasks. This allows the use of either a top-down or a bottom-up methodology. If the programmer chooses to create an overview of the system and work down, the system will keep up with unfinished boxes and display them in red or yellow. A library routine is specified by naming a box, allowing the system to insert the necessary code if it already exists and to display the box in green. A history will be maintained to allow the programmer/engineer to backtrack from undesired refinements. This file could also be used to display top-down structure graphs of the system being developed and its state of completion. A system with these features is currently under development.

2.6 A Sample Program

As a simple example of the kinds of problems encountered in the world of process control, suppose we have some exothermic chemical reaction that takes place in a chamber. A coolant is available to keep the process from overheating, and the task is to control the valve that allows the coolant to flow through the chamber. At the same time, an operator's display that indicates the chamber temperature is to be updated every second. A temperature sensor (in the form of an A/D converter) provides this reading, and the control algorithm is simply to open the coolant valve whenever the temperature exceeds 85 degrees, and to close it again whenever it falls below 82.

Clearly, ordinary programming languages will not allow a solution to this problem, due in part to their inability to cope with the requirement of updating the operator's display once a second. The code necessary to access the A/D converter and the operator's display could be written in some high order languages, or directly in assembly language. However, dealing with the real-time constraint would involve a substantial programming effort, including perhaps a scheduler and some routines to manipulate a real-time clock.

Figure 3 provides a simple real-time data-flow solution to the scheduling problem. The approach to creating this graph is to identify first the devices that must be accessed (the temperature measurement, the display, and the coolant control) and to create a box to operate each. In this case, each is a sequential code box. The temperature measurement box takes any data token as an indication that sampling should take place. This box produces a token having the value of the temperature sampled. At this stage, the box is truly a "black box" (actually, red on the color display) with one input and one output. Similarly, the display box has a single input, the value to be displayed, and produces no output. Of course, it does have some effect, namely updating the operator's display. The third box is somewhat more complicated, but still simple in absolute terms. If it receives a true token, it should open the coolant

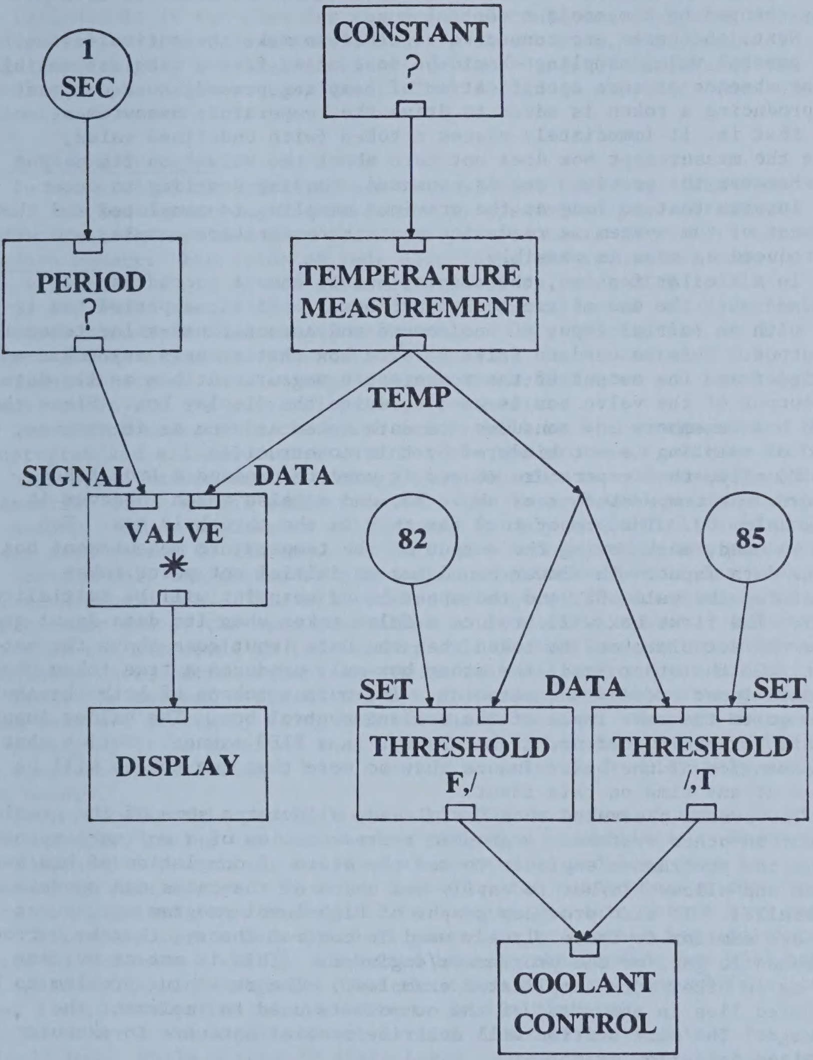

1
SEC
CONSTANT
?
PERIOD
?
TEMPERATURE
MEASUREMENT
TEMP
SIGNAL
DATA
VALVE
*
DISPLAY
82
85
SET
DATA
SET
THRESHOLD
F,/
THRESHOLD
/,T
COOLANT
CONTROL

FIGURE 3

valve, and if it receives a false token, it should close the coolant valve. Presumably this is by means of a switch or a D/A converter, but at this level of the design that is irrelevant. The assumption is only that the coolant valve remains either open or closed until it is explicitly changed by the coolant control box.

Next, the boxes are connected together to make the entire system. As a general rule, sampling should be done at as fast a rate as possible, in the absence of some specification of sampling period, so a constant box producing a token is added to drive the temperature measurement box; that is, it immediately places a token (with undefined value, since the measurement box does not care about the value) on its output arc whenever the previous one is consumed, causing sampling to occur. This insures that so long as the previous sampling is completed and that the rest of the system is ready for another temperature sample, one will be produced as soon as possible.

In a similar fashion, the display update once a second can be obtained with the use of two additional boxes. First, a period box is used with an initial input of one second and an undefined-value token to be output. This is used to drive a valve box that accepts any token as a trigger and the output of the temperature measurement box as its data. The output of the valve box is used to drive the display box. Since the valve box remembers and consumes its data token as soon as it arrives, the speed of sampling is not hindered by this construction.

Finally, the temperature sensed is used to produce a true token whenever the temperature goes above 85, and a false token whenever it falls below 82. The proper tool for this is the threshold box. Two will be used, each having the output of the temperature measurement box as its data input. The lower bound has an initial set point token containing the value 82, and the upper bound setpoint will be initialized to 85. The first box will produce a false token when its data input goes below the setpoint, and no token when the data input goes above the setpoint. On the other hand, the other box only produces a true token when the data input exceeds the setpoint. The output tokens of both threshold boxes go to the same input of the coolant control box. The values input to this box are queued up, if necessary, in a FIFO manner. Notice that the semantics of the boxes insure that no more than two tokens will be queued at any time on this input.

The system presented thus far already alleviates some of the problems present in other systems. A graphic representation of a software system allows the programmer/engineer to see the state of completion of his/her system and allows him/her to easily see which of the tasks can operate in parallel. It also provides graphs of high-level program structures that are similar to those already used in control theory, thereby narrowing the semantic gap for the programmer/engineer. (This is not as evident here as in other more complicated examples.) The remaining problem to be conquered lies in the speed of the components used to implement the language. The next section will describe special hardware to execute real-time data-flow graphs.

3. A PROCESSOR TO EXECUTE THE LANGUAGE

The direct compilation of the graph in Figure 3 into executable code for an ordinary vonNeumann machine is a formidable but not impossible task. Essentially, the graph can be represented internally by a fairly complicated data structure, accompanied by an interpreter that can determine when the boxes are ready to fire and schedule them in some manner on the single processor machine. This approach has one advantage: portability. In theory, only this interpreter, the run-time support, and perhaps some device-specific code in the boxes, would need to be changed to move to another system. With the situation in the microprocessor industry being that the processor ordered today is obsolete by the time that the machine is delivered, portability is a major concern in practice.

On the other hand, this approach fails where most other attempts have failed: it is too slow for the required real-time responses demanded by many applications. As a result of this shortcoming, a preliminary design for a highly parallel architecture that directly implements the data-flow graphs has been completed. That design is the basis for this section.

3.1 A System Overview

In our preliminary architecture to implement real-time data-flow graphs, the box operations are performed by special-purpose parallel hardware that is provided on a processor-per-box basis in what is termed the graph memory. Execution of code describing the sequential code boxes will be by a set of conventional processors, each with local memory, and whose number impacts only the potential speed of sequential code boxes when more than one is executable. Only descriptors of the sequential code boxes and information regarding their readiness for execution are stored in the graph memory; the actual code and local data storage are in the conventional processor memories.

The box processors in the graph memory will be microprogrammed. The microprogram and all associated data for the box processor will be stored in a high-speed random access memory, that can also be addressed by other processors. Under normal conditions only the "owner" will be accessing this memory. Other processors will access it only to initialize it and to pass data tokens to the box. As mentioned before, these box processors will contain only the necessary microcode to perform a particular task, e.g., a threshold box. This will limit the necessary address space of a box processor and at the same time allow an extensible system. Information other than the microcode in this memory will include connection information to indicate where output otkens are to be sent and where input tokens come from, the current state of the box processor, and data storage for tokens that are currently being held in the processor. Through the use of VLSI packaging it is expected that several box processors may be built on a single chip, allowing a high density to exist in a relatively small graph memory.

Theoretically this parallelism would allow a number of boxes to fire simultaneously, giving the system nice real-time capabilities. However, as was pointed out earlier, a number of contention problems must be addressed in this system. By using data-flow the contention on the shared memory is minimized since boxes will be waiting for data before they can fire. Likewise, once a data token is passed on to another box additional updates to the memory will be minimized because the sending box will not fire again until it is signaled that the token has been removed from its output link. Contention is, in a sense, self-minimizing.

The other problem pointed out earlier dealt with contention for a single bus. While a totally distributed system is out of the question because of synchronization problems, the box processors can be somewhat autonomous. The bus structure chosen to provide the communications of the system is discussed in the next subsection.

3.2 A Solution to the Bus Bottleneck

A hierarchical bus structure has been selected for this system (see Figure 4). It consists of all of the processors, both box and conventional, and a set of busses and their related controllers. At the lowest level each box processor will have total reign over its own bus. The next highest level would be between the box processors on a single chip.

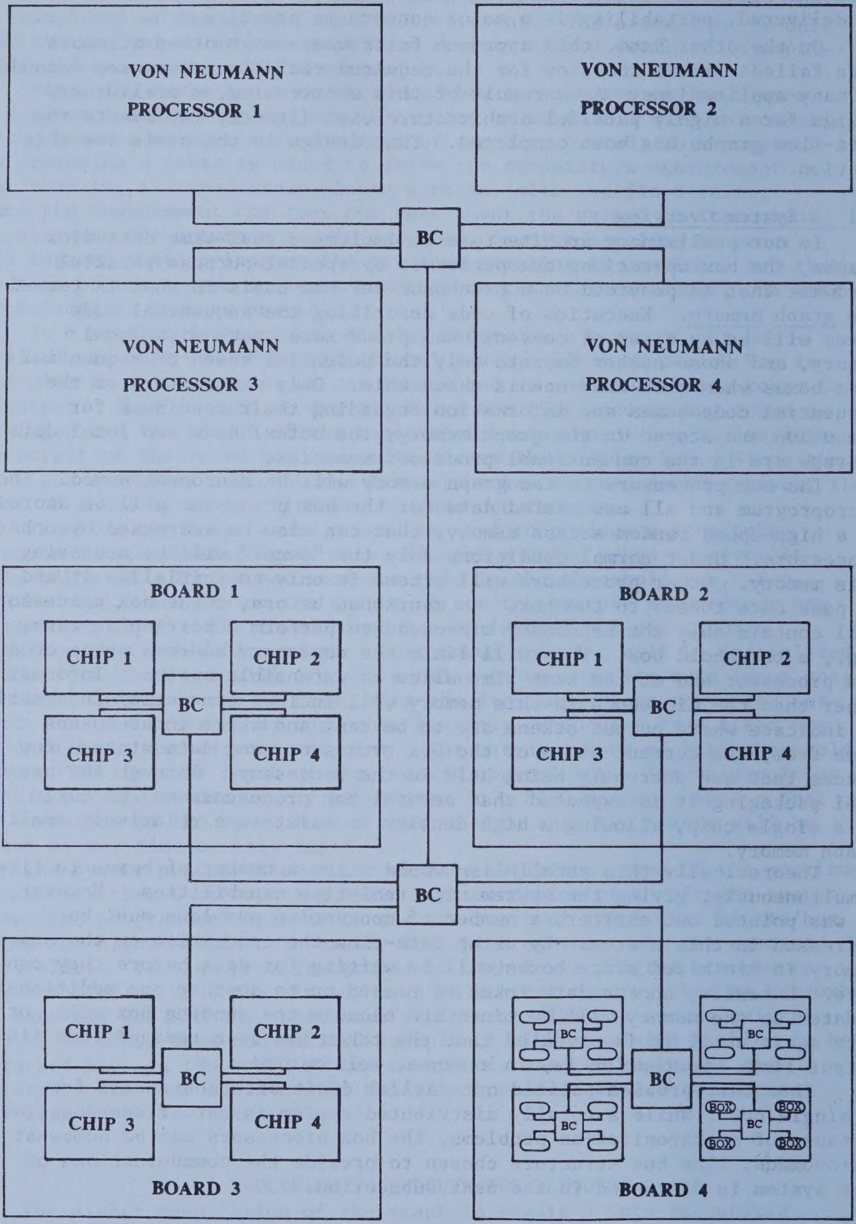

VON NEUMANN PROCESSOR 1
VON NEUMANN PROCESSOR 2
BC
VON NEUMANN PROCESSOR 3
VON NEUMANN PROCESSOR 4
BOARD 1
CHIP 1
CHIP 2
BC
CHIP 3
CHIP 4
BOARD 2
CHIP 1
CHIP 2
BC
CHIP 3
CHIP 4
BC
CHIP 1
CHIP 2
BC
CHIP 3
CHIP 4
BOARD 3
BC
BC
BC
BC
BOX
BOX
BOX
BOX
BC
BOARD 4

FIGURE 4

Communication from one processor to another on a chip would not affect any other processors off the chip. At the next level up, the board level, processors trying to contact other processors off the chip must do so at both the chip level and the board level. Any processors that are not trying to communicate at the board level will be able to do so, and any processor wishing to address only its own bus will not be delayed. Up another level is the inter-board level, which allows processors on different boards to communicate with each other. Finally, there would be a processor/graph memory bus to allow box processors to communicate with the conventional processors.

All communications between levels will be handled by the bus controllers. The individual processors will only know that they are accessing an address in some memory. The address will be broken down into fields, with a few bits representing the board number, a few representing the chip, a few the box, etc. The normal handshaking on these accesses will be minimized due to the data-flow aspects of the system; it is necessary only to write to a remote memory, never to read from one. This structure is also nice in that it can be shown that there is an efficient algorithm for finding the optimum mapping of processes onto a tree structure that will yield a minimum of communication conflict [8].

4. CONCLUSION

The real-time data-flow architecture provides solutions to many of the problems faced in process control by narrowing the semantic gap between language and user, and providing an easy yet efficient processing system lacking the contention problems found in systems in use today. The architecture is designed to allow ease of expansion from both the hardware and software standpoints. Future research will include production of software systems, simulations and analysis to optimize the system, and more detailed design of the individual components of the hardware, eventually leading to a prototype system.

REFERENCES

[1] Allichin, J. Modula and a Question of Time. IEEE Transactions on Software Engineering SE-6, 4 (July 1980), pp. 390-391.

[2] Brinch Hansen, P. The Architecture of Concurrent Programs. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1977.

[3] Dennis, J. B. First Version of a Data-Flow Procedure Language. Lecture Notes in Computer Science 19, Springer-Verlag, Heildelberg, 1974, pp. 362-376.

[4] Dennis, J. B. and Misunas, D. P. A Preliminary Architecture for a Basic Data-Flow Processor. SIGARCH News 3 (Proc. 2nd Annual Symposium on Computer Architecture), 4 (Dec. 1974), pp. 126-132.

[5] Lamport, L. Time, Clocks, and the Ordering of Events in a Distributed System. Communications of the ACM 21, 7 (July 1978), pp. 558-565.

[6] Wegner, P. Programming with ADA: An Introduction by Means of Graduated Examples. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1979.

[7] Wirth, N. Modula: A Language for Modular Multiprogramming. Software Practice and Experience 2, 1 (Jan.-Feb. 1977), pp. 3-36.

[8] Wu, S. B. and Liu, M. T. Assignment of Tasks and Resources for Distributed Processing. Proc. COMPCON 80, Sept. 1980, pp. 655-662.

PERMUTING WITH THE AUGMENTED DATA MANIPULATOR NETWORK

ROBERT J. MCMILLEN
GEORGE B. ADAMS III
HOWARD JAY SIEGEL

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

The problem of how to control the augmented data manipulator (ADM) and the inverse ADM (IADM) SIMD machine interconnection networks is explored. To distribute control of the network among the processors of the system, routing tags are investigated. It is proven that positive (or negative) dominant tags can be used to control the network to pass two of Lenfant's five permutation classes. An algorithm to determine whether an arbitrary permutation is passable without blocking using dominant or natural routing tags is presented.

I. INTRODUCTION

In [7,12], it is shown that for SIMD processing the capabilities of the augmented data manipulator (ADM) network are a superset of those of multistage cube networks. Various properties of the ADM have been discussed in [9,12-14]. In [5] a routing tag scheme was developed to allow the ADM network to be controlled in a distributed fashion in an MIMD (multiple instruction stream - multiple data stream) [2] environment. In this paper, the capabilities of an ADM routing tag scheme operating in an SIMD environment are studied. The abilities of the cube-type multistage networks and the ADM/IADM networks are being compared in order to choose a network for PASM, a partitionable SIMD/MIMD multimicroprocessor system being designed at Purdue [10,11].

The augmented data manipulator and inverse ADM (IADM) networks are defined in Section II. In Section III the routing scheme and several methods for generating routing tags are presented. The ability of the routing tags to route some classes of permutations without conflict is shown. Section IV presents an algorithm for determining if an arbitrary permutation can be routed by "dominant" or "natural" tags.

II. NETWORK DEFINITION

The model for an SIMD (single instruction stream - multiple data stream) machine architecture [2] used here includes a control unit which broadcasts instructions to N processors and an interconnection network to allow communication among the processors [8]. A processor along with its private memory is called a processing element or PE. All active PEs execute the same instruction at the same time, each processor on data from its own memory. Data is transferred by the interconnection network from PE to PE, and so this model is called the PE-to-PE model. Each PE is assigned a unique address from 0 to N-1, where $N=2^n$.

This work was sponsored by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant No. AFOSR-78-3581. The United States Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

An interconnection network can be described as a set of interconnection functions, where each interconnection function is a permutation (bijection) on the set of PE addresses [6]. When interconnection function f is applied, input i is connected to f(i) for all i, $0 \le i < N$, simultaneously.

The Plus-Minus 2^i (PM2I) network consists of the following functions:

$$PM2_{+i}(j) = j + 2^i \bmod N \text{ and } PM2_{-i}(j) = j - 2^i \bmod N$$

for $0 \le j < N$, $0 \le i < n$ [6], where $-x \bmod N = N-x \bmod N$. The data manipulator network [1], shown in Figure 1, consists of n stages, with N switching cells per stage, and a column of network output cells. The stages are ordered from n-1 to 0, where the interconnection functions of stage i are $PM2_{+i}$, $PM2_{-i}$, and the identity (straight across). The controls of the data manipulator are limited to one pair per stage. The augmented data manipulator (ADM) [7,12] is a data manipulator with individual cell control, i.e., each cell can be set independently of the other cells. The data output from cell P at stage i becomes the data input to cell P' at stage i-1, where $P' \in \{P, (P+2^i) \bmod N, (P-2^i) \bmod N\}$. If the stages of the ADM network are traversed in reverse order, the resulting network is called the inverse augmented data manipulator (IADM).

III. NETWORK CONTROL

To distribute control of the network among the N PEs, routing tags are used. To be able to specify any arbitrary path in the network, a full routing tag is required at each network input. The full tag requires 2n bits and is of the form $F = (f_{2n-1} f_{2n-2} \ldots f_1 f_0)$. The even numbered bits

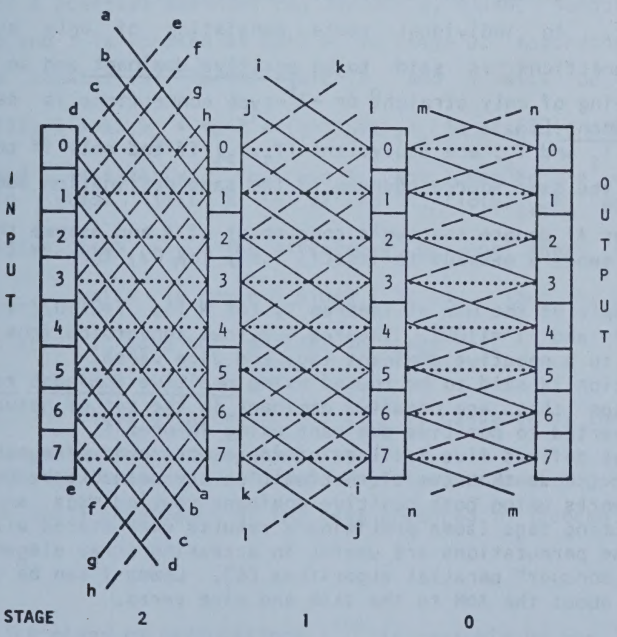


Figure 1: The Data Manipulator for N=8. The lower case letters represent "end-around" connections.

represent the magnitude of the route and the odd numbered bits represent the signs corresponding to the magnitude bits. In stage i, a given switching element examines bits 2i and 2i+1 of the full tag. If $f_{2i} = 0$, the straight link is used regardless of the value of f_{2i+1}. If $f_{2i} = 1$ and $f_{2i+1} = 0$, the $+2^i$ link is used. If $f_{2i} = 1$ and $f_{2i+1} = 1$, the -2^i link is used. For example in the IADM network, for N=16, if the source is 3 and the destination is 10, one possible value for F is 01001101. The path traversed is $+2^0$, -2^1, straight, $+2^3$.

If all the sign bits in a full tag are the same, the information contained in those bits can be represented by one bit which is the sign bit for the whole tag. An n+1 bit routing tag can be formed by computing the signed magnitude difference between the destination, D, and the source, S: $T = (t_n t_{n-1} \ldots t_1 t_0) = D - S$. The sign bit is t_n, where $t_n = 0$ indicates positive and $t_n = 1$ indicates negative. Bits $t_{n-1} \ldots t_1 t_0$ equal the absolute value of D − S, i.e., the magnitude of the difference [5]. If all N routing tags for a permutation are calculated in this way, then the permutation is said to be routed using natural routing tags. For example, if N = 16, S = 1011, and D = 0100, then T = 10111.

To route a message through the network, stage i need only examine bits t_n and t_i in the routing tag. If $t_i = 0$, the straight connection is used regardless of the value of t_n. If $t_n = 0$ and $t_i = 1$, the $+2^i$ link is used, and if $t_n = 1$ and $t_i = 1$, the -2^i link is used. In the previous example, with T = 10111, if a message enters the IADM network at stage 0, the sequence of connections traversed from processor 11 to 4 is -2^0, -2^1, -2^2, straight. An individual route consisting of only straight or $+2^i$-type connections is said to be positive dominant and an individual route consisting of only straight or -2^i-type connections is said to be negative dominant [5].

Two tags, T_1 and T_2, are equivalent ($T_1 \approx T_2$) if and only if they route a message from the same source address to the same destination address.

Theorem 1: Let A' denote the two's complement of A and assume that a processor never sends a message to itself, i.e., $T \neq 0$. Then $T' \approx T$ [5].

[]

As an example of the use of Theorem 1, let N=16, S=0110, and D=1101, then T=00111 and T'=11001. Theorem 1 allows conversion from a positive dominant tag to a negative dominant tag, and vice versa.

A permutation is said to be routed using positive dominant routing tags if those tags that are negative dominant in the set of natural routing tags are converted to positive dominant using Theorem 1.

Lenfant has defined five families of frequently used permutations [4]. This section shows that two of the families are passable by both the ADM and IADM networks using both positive dominant routing tags and negative dominant routing tags (some preliminary results were stated without proof in [14]). The permutations are useful in accessing array elements and in "divide and conquer" parallel algorithms [4]. Lemma 1 can be used to extend results about the ADM to the IADM and vice versa.

Lemma 1: The ADM passes the permutation f if any only if the IADM passes the inverse permutation [14].

[]

The following notation is used in the theorems: $(p_{n-1}\cdots p_1p_0) = (\hat{a}_2,\hat{a}_1)$ is the bitwise representation of an address P, $\hat{a}_2 = (p_{n-1}p_{n-2}\cdots p_{n-j})$, the j high order bits of P, and $\hat{a}_1 = (p_{n-j-1}\cdots p_1p_0)$, the n-j low order bits of P. $<T>_R$ denotes T mod 2^R.

Lemma 2: The location of a message in stage i of the IADM network is cell $<S+<T>_i>_n$, where S is the source address and T is the magnitude portion of a positive dominant routing tag for the message.

Proof: The routing tag is interpreted one bit at a time in the network beginning with the low order bit. At stage i, bits 0 through i-1 have been examined, thus the message has been displaced by $<T>_i$.

[]

Theorem 2: The class of permutations $\lambda_{j,k}^{(n)}$, defined as mapping X to jX+k mod 2^n (j odd), is passable by the IADM network using positive dominant routing tags.

Proof: For a conflict to occur between two messages currently in stage i, using positive dominant tags, they must be a distance of 2^i apart. The conflict occurs at the input to stage i+1 when both messages try to gain access to the same cell. Because of the nature of this class of permutations, if two messages are 2^i apart in stage i, they were 2^i apart in stage 0, where they started. Examine the routing tags messages 2^i apart in stage 0 use, and where the tags move them to in stage i. By definition, the tag is $T_X = <<jX+k>_n - X>_n = <(j-1)X+k>_n$, which specifies the magnitude of a positive dominant tag for all X, $0\leq X<N$. Suppose X is located at p and Y is located at $<p+2^i>_n$ in stage 0. According to Lemma 2, X will be located at $<p+<(j-1)p+k>_i>_n$ and Y will be located at $<p+2^i+<(j-1)(p+2^i)+k>_i>_n = <p+2^i+<(j-1)p+k>_i>_n$ in stage i. The difference between X and Y is clearly 2^i in stage i. Now, compute the location of X and Y in stage i+1: X is at $<p+<(j-1)p+k>_{i+1}>_n$ and Y is at $<p+2^i+<(j-1)(p+2^i)+k>_{i+1}>_n$. The difference between these locations is $2^i+<(j-1)2^i>_{i+1}$, which is never 0 since j is odd. Thus any pair X and Y, 2^i apart in stage i, does not conflict in stage i+1.

[]

Corollary 1: The class of permutations $\lambda_{j,k}^{(n)}$ is passable by the IADM network using negative dominant routing tags.

Proof: Similar to Theorem 2. (To find negative dominant tags for X and Y, compute the two's complement of the tags computed in Theorem 2 (e.g. $T_X' = <N-(j-1)X-k>_n$).)

[]

Theorem 3: The class of permutations $\lambda_{j,k}^{(n)}$ is passable by the ADM network in one pass using positive dominant routing tags or negative dominant routing tags.

Proof: Assume all arithmetic is mod N. $\lambda^{(n)-1}_{j,k}$ maps jX+k to X, but that is equivalent to jX → X+k', ("→" means "maps to") where k' is the additive inverse of k in the ring of integers mod N. Further, if j' is the multiplicative inverse of j in the ring of integers mod N then the mapping becomes X → j'X + j'k'. Since j is odd, N is a power of two, j and N are mutually prime, and arithmetic is mod N, j' exists and is an odd integer [3]. Since $\lambda^{(n)-1}_{j,k} = \lambda^{(n)}_{j',j'k'}$ it is of the class $\lambda^{(n)}_{j,k}$. The rest follows from Lemma 1, Theorem 2, and Corollary 1.

[]

Theorem 4: The class of permutations $\delta^{(n)}_{j,k}$, defined as mapping $(\hat{a}_2,\hat{a}_1)$ to $(\hat{a}_2,\lambda^{(n-j)}_{1,k}(\hat{a}_1))$ ($j \le n$), is passable by the IADM network using positive dominant routing tags.

Proof: The source can be represented as $\hat{a}_2 2^{n-j}+\hat{a}_1$ and the desired destination as $\hat{a}_2 2^{n-j}+\langle \hat{a}_1+k\rangle_{n-j}$. By definition, then the positive dominant routing tag is $+\langle\langle \hat{a}_1+k\rangle_{n-j}-\hat{a}_1\rangle_n$. Without loss of generality, assume $0 \le k < 2^{n-j}$ since any arbitrary k can be mapped to this range by taking the modulus 2^{n-j}. If $\hat{a}_1 < 2^{n-j}-k$, then the routing tag is simply +k or $(00\ldots0k_{n-j-1}\ldots k_1k_0)$. If $\hat{a}_1 \ge 2^{n-j}-k$, $\langle \hat{a}_1+k\rangle_{n-j} = \hat{a}_1+k-2^{n-j}$. The routing tag becomes $T = \langle \hat{a}_1+k-2^{n-j}-\hat{a}_1\rangle_n = \langle k-2^{n-j}\rangle_n$. Since $k<2^{n-j}$, $T = 2^n-2^{n-j}+k = (2^j-1)2^{n-j}+k$. Represented as bits, $T = (11\ldots1k_{n-j-1}\ldots k_1k_0)$, thus

$$T = \begin{cases} (00\ldots0k_{n-j-1}\ldots k_1k_0) & \hat{a}_1 < 2^{n-j}-k \\ (11\ldots1k_{n-j-1}\ldots k_1k_0) & \hat{a}_1 \ge 2^{n-j}-k \end{cases}$$

There cannot be any conflict in stages 0 through n-j-1 since all routing tags are the same in bits 0 through n-j-1. For two messages to conflict in stage n-j, they must be 2^{n-j} apart in stage n-j-1 (since positive dominant tags are being used). All messages in stage n-j-1 are a distance of k from their original location, so the relative distances between messages does not change. Messages in stage 0 that are a distance of 2^{n-j} apart will be using the same routing tag, since their $\hat{a}_1$'s will be the same, and thus will not conflict in stage n-j. In general, messages in stage 0 that are a distance of 2^p apart, $n-j \le p < n$ will not conflict in stage p because they are using the same routing tag.

[]

Corollary 2: The class of permutations $\delta^{(n)}_{j,k}$ is passable by the IADM network using negative dominant routing tags.

Proof: Since $\lambda^{(n)}_{1,k}$ is the identity mapping for k=0, assume without loss of generality that k≠0. Let the two's complement of k, k', be $k'_{n-j}\ldots k'_1k'_0$. To find the negative dominant tags for the permutation, compute the two's complement of the tags calculated in Theorem 4:

$$T = \begin{cases} (11\ldots1k'_{n-j-1}\ldots k'_1k'_0) & \hat{a} < 2^{n-j-1}-k \\ (00\ldots0k'_{n-j-1}\ldots k'_1k'_0) & \hat{a}_1 \geq 2^{n-j-1}-k. \end{cases}$$

The rest follows from the same reasoning used in Theorem 4.

[]

Theorem 5: The class of permutations $\delta^{(n)}_{j,k}$ $(j \leq n)$ is passable by the ADM network using positive dominant routing tags or negative dominant routing tags.

Proof: $\delta^{(n)-1}_{j,k}$ maps $(\hat{a}_2, \lambda^{(n-j)}_{1,k}(\hat{a}_1))$ to $(\hat{a}_2, \hat{a}_1)$. This is equivalent to mapping $(\hat{a}_2, \hat{a}_1)$ to $(\hat{a}_2, \lambda^{(n-j)-1}_{1,k}(\hat{a}_1))$. From Theorem 3, λ^{-1} is the same class as λ, thus δ^{-1} is in the same class as δ. The rest follows from Lemma 1, Theorem 4, and Corollary 2.

[]

Positive dominant routing tags cannot be used to route all passable permutations without conflict. One such permutation is the perfect shuffle [15] which maps $(p_{n-1}\ldots p_1p_0)$ to $(p_{n-2}\ldots p_1p_0p_{n-1})$ for all $0 \leq P < N$. The shuffle can be routed without conflict using natural tags as shown in Theorem 6. In the theorem, use is made of a result obtained in [14]. An algorithm is presented there which calculates the switch settings at each stage, so that the ADM network can pass the perfect shuffle. The switch setting algorithm is presented here for easy reference:

The switch settings for stage i, $n > i \geq 0$, are determined as follows, where the address of a cell P at stage i is $p_{n-1}\ldots p_1p_0$.

```
set stage n-1 to straight across;
for i = n-2 step -1 until 0
    if $p_{i+1} \neq p_i$ then if $p_{i+1} = 0$
                          then set cell P at stage i to $PM2_{+i}$;
                          else set cell P at stage i to $PM2_{-i}$;
                  else set stage i to straight across;
```

Theorem 6: The perfect shuffle is passable by the ADM network using natural routing tags.

Proof:

Case 1: $p_{n-1} = 1$. Then $P \geq \text{shuffle}(P)$. The equality is true when the p_i are all 1. When there is at least one $p_i = 0$ $(i \neq n-1)$, consider that which is most significant. The most significant 0 in shuffle(P) will be in the i+1st position. Since there is guaranteed to be a 1 in the i+1st and higher positions in P, $P > \text{shuffle}(P)$. Since shuffle(P) is the destination address of source P, $T = (\text{shuffle}(P)-P) \leq 0$, which implies T is negative dominant. Now examine the switch setting algorithm. If $p_{n-1} = 1$, then the bit pair $p_{i+1}p_i$ will always be of the form 10 or 11. The algorithm specifies settings of -2^i and straight respectively. Thus the negative dominant tag specifies the correct settings.

Case 2: $p_{n-1} = 0$. By similar reasoning, $P \leq \text{shuffle}(P)$, where the equality occurs if all p_i are 0. $T \geq 0$ which implies a positive dominant routing tag. In the switch setting algorithm, the bit pair $p_{i+1}p_i$ will always be either 00 or 01, specifying routes of straight or $+2^i$ respectively.

[]

Corollary 3: The inverse shuffle permutation is passable by the IADM network using natural routing tags.

Proof: Follows from Lemma 1 and Theorem 6.

[]

The routing tags used in this section require only n+1 bits and are easy to compute. Thus the overhead required for routing is reduced when they can be used. If a known passable permutation is frequently used or an important part of a computation, but it cannot be represented with natural or positive dominant routing tags, full routing tags can always be precomputed.

IV. DETERMINING PASSABILITY

To analyze the passability of a particular permutation on the ADM using positive dominant routing tags, the algorithm below begins by comparing the least significant bit of a source/destination pair. If these bits are the same, the destination value, dest(j), of source j is not changed. This corresponds to a straight connection in stage 0. If they differ, dest(j) is revised by subtracting 2^0. This corresponds to the PM2I connections of stage 0 used by positive dominant tags.

At this point the algorithm checks for possible conflict with other stage 0 connections requested by other revised dest values. If a conflict is found the permutation is declared not passable (in a single pass) and the algorithm terminates. If there is no conflict, the algorithm proceeds with the remaining source/destination pairs, checking for conflict.

At this point, if there have been no conflicts each original destination has been updated, although its value may be unchanged. The process begins anew with stage 1, this time comparing the next least significant bits of each source/updated-destination pair.

The process is repeated in successive stages until either a conflict is detected or each stage 0 through n-2 has been checked. Stage n-1 need not be checked.

Consider the case where the permutation under test by the algorithm has been found free from conflict from stage 0 to n-2. In stage n-1 the physical structure of the network only allows a node with address A to be connected to the node with the same address or the node with address $A+2^{n-1}$ mod N, and vice versa. Since the network contains positive dominant paths from A to A, A to $A+2^{n-1}$ mod N, and $A+2^{n-1}$ mod N to A, any connection which can be performed in stage n-1 can be specified by positive dominant routing tags. Therefore, conflict cannot occur in stage n-1, so the algorithm does not check for it.

First an algorithm is given to check for passability with positive dominant tags. The algorithm is then modified to check for passability with negative dominant tags, and then for natural routing tags. Operation with negative dominant and natural tags is similar that with positive dominant tags.

The process to check the passability of a given permutation using positive dominant routing tags can be described by the procedure in Figure 2. In the procedure, dest is an N-element vector where dest(j) is the destination of source j. The notation X_i is used to denote bit i of X.

The procedure REQUEST, called in procedure PASSABILITY, is the portion of the algorithm which generates the needed connections based on the type of routing tags used. This is the only aspect of the algorithm that is

```
procedure PASSABILITY(dest,N);
passable = true;
while passable = true do
        for i = 0 step 1 until $\log_2 N-2$ do
                if passable = false then stop;
                check = 0; /*check is an N-element bit vector*/
                        for j=0 step 1 until N-1 do
                                if $dest(j)_i \neq j_i$
                                        then REQUEST(dest(j),i,j);
                        if check(dest(j)) = 0
                                then check(dest(j))=1;
                                else passable=false;
```

Figure 2: Procedure PASSABILITY for testing permutation passability.

dependent on the type of routing tags used. Three versions of procedure REQUEST are given, one each for positive dominant, negative dominant, and natural routing tags.

For positive dominant routing tags, procedure REQUEST is:

$$dest(j)=(dest(j)-2^i) \bmod N.$$

For negative dominant routing tags, procedure REQUEST is:

$$dest(j)=(dest(j)+2^i) \bmod N.$$

For natural routing tags, procedure REQUEST is:

```
if $(dest(j)-j) > 0$ then $dest(j)=(dest(j)-2^i) \bmod N$
                          else $dest(j)=(dest(j)+2^i) \bmod N$
```

The algorithm can be shown to give a correct result by demonstrating that it imitates the subtraction performed to calculate a routing tag. First consider the case where positive dominant tags are desired and D is a particular destination and S its source.

Theorem 7: Let D be a particular destination, S its source, and $R^{(i)}$ be the revised value of D after the inner loop of the algorithm has been successfully completed i times. $X_{s/t}$ represents bits s through t, inclusive, of X. Then $R^{(i)}_{n-1/i+1} = (D-S_{i/0})_{n-1/i+1}$, where $0 \leq i \leq n-2$.

Proof: Basis step: i = 0. $R^{(0)}_{n-1/1} = (D-S_{0/0})_{n-1/1}$ by definition.

Induction step: Assume true for i, show true for i+1. Case $S_{i+1} = R^{(i)}_{i+1}$: $R^{(i+1)} = R^{(i)}$. Therefore, $R^{(i+1)}_{n-1/i+2} = R^{(i)}_{n-1/i+2} = (D-S_{i/0})_{n-1/i+2} = (D-S_{i+1/0})_{n-1/i+2}$ (since $S_{i+1} = R^{(i)}_{i+1} = (D-S_{i/0})_{i+1}$ no borrow can propagate into the i+2 bit position). Case $S_{i+1} \neq R^{(i)}_{i+1}$: when $S_{i+1} \neq R^{(i)}_{i+1}$ the algorithm subtracts 2^{i+1}. Subcase $S_i = 0$, $R^{(i)}_{i+1} = 1$. $R^{(i+1)}_{n-1/i+2} = (R^{(i)}_{n-1/i+1} - 2^{i+1})_{n-1/i+2} = (R^{(i)}_{n-1/i+2} + 2^{i+1} - 2^{i+1})_{n-1/i+2} = R^{(i)}_{n-1/i+2} =$

$(D-S_{i/0})_{n-1/i+2} = (D-S_{i+1/0})_{n-1/i+2}$ (since $S_{i+1} = 0$). Subcase $S_{i+1} = 1$, $R^{(i)}_{i+1} = 0$. $R^{(i+1)}_{n-1/i+2} = (R^{(i)}_{n-1/i+1} - 2^{i+1})_{n-1/i+2} =$ $((D-S_{i/0})_{n-1/i+1} - 2^{i+1})_{n-1/i+2} = ((D-S_{i+1/0})_{n-1/i+1})_{n-1/i+2} =$ $(D-S_{i+1/0})_{n-1/i+2}$.

[]

For positive dominant tags: $|T| = D-S$ when $D>S$, and $|T| = N-(S-D) = N+(D-S)$ when $S>D$ (since two's complement of $|D-\overline{S}| = N-(S-D)$). Thus, $T_{i+1} = (D-S_{i/0})_{i+1} - S_{i+1}$. From this, $T_{i+1} = (D-S_{i/0})_{i+1} \oplus S_{i+1}$. From Theorem 7, $(D-S_{i/0})_{i+1} = R^{(i)}_{i+1}$. Therefore, when the algorithm compares $R^{(i)}_{i+1}$ and S_{i+1} the correct path is chosen. Thus the algorithm faithfully simulates the operation of the positive dominant routing tag. If negative dominant tags are chosen a similar argument shows that the algorithm simulates this situation accurately. Since both positive and negative dominant routing tags are handled correctly, the algorithm can operate correctly with natural routing tags.

The time complexity of the algorithm is $O(N\log_2 N)$, in all cases. The space complexity is $O(N)$, in particular, one vector of length N.

To adapt the algorithm to use with the IADM, exchanging all plus and minus signs in REQUEST and using a source vector instead of a destination vector will result in the correct operation of the algorithm. This is because the IADM network will be checked for conflict from input stage toward output stage, the opposite of the ADM.

V. CONCLUSIONS

The capabilities of a routing tag scheme developed for the ADM and IADM networks operated in an SIMD environment were explored. Several types of permutation routing tags were defined. Full permutation routing tags are capable of routing any permutation known to be passable by the ADM and IADM networks. It was shown that positive and negative dominant permutation routing tags, which are easily calculated, are capable of routing two classes of useful permutations defined by Lenfant in the ADM and IADM networks. Natural permutation routing tags, also easily calculated, are capable of routing the perfect shuffle permutation in the ADM network and the inverse perfect shuffle permutation in the IADM network. Finally, an algorithm was presented that determined if a given arbitrary permutation can be routed by the ADM or IADM network given positive dominant, negative dominant, or natural permutation routing tags.

VI. REFERENCES

[1] T. Feng, "Data manipulating functions in parallel processors and their implementations," *IEEE Trans. Comp.*, vol. C-23, Mar. 1974, pp. 309-318.

[2] M. Flynn, "Very high-speed computing systems," *Proc. IEEE*, vol. 54, no. 12, Dec. 1966, pp. 1901-1909.

[3] L. Goldstein, *Abstract Algebra: A First Course*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1973.

[4] J. Lenfant, "Parallel permutations of data: A Benes network control algorithm for frequently used permutations," *IEEE Trans. Comp.*, vol. C-27, July 1978, pp. 637-647.

[5] R.J. McMillen and H.J. Siegel, "MIMD machine communication using the

augmented data manipulator network," 7th Symp. Comp. Arch., June 1980, pp. 51-58.

[6] H.J. Siegel, "Analysis techniques for SIMD machine interconnection networks and the effects of processor address masks," IEEE Trans. Comp., vol. C-26, Feb. 1977, pp. 153-161.

[7] H.J. Siegel, "Interconnection networks for SIMD machines," Computer, vol. 12, June 1979, pp. 57-65.

[8] H.J. Siegel, "A model of SIMD machines and a comparison of various interconnection networks," IEEE Trans. Comp., vol. C-28, Dec. 1979, pp. 907-917.

[9] H.J. Siegel, "The theory underlying the partitioning of permutation networks," IEEE Trans. Comp., vol. C-29, Sept. 1980, pp. 791-801.

[10] H.J. Siegel, P. Mueller, H. Smalley, "Control of a partitionable multimicroprocessor system," 1978 Int'l. Conf. Parallel Processing, Aug. 1978, pp. 9-17.

[11] H.J. Siegel, L.J. Siegel, R.J. McMillen, P. Mueller, S. Smith, "An SIMD/MIMD multimicroprocessor system for image processing and image processing," IEEE Comp. Soc. Conf. Pat. Recog. and Image Proc., Aug. 1979, pp. 214-224.

[12] H.J. Siegel and S. Smith, "Study of multistage SIMD interconnection networks," 5th Symp. Comp. Arch., Apr. 1978, pp. 223-229.

[13] S. Smith and H.J. Siegel, "Recirculating, pipelined, and multistage SIMD interconnection networks," 1978 Int'l. Conf. Parallel Processing, Aug. 1978, pp. 206-214.

[14] S. Smith, H.J. Siegel, R.J. McMillen, G.B. Adams III, "Use of the augmented data manipulator multistage network for SIMD machines," 1980 Int'l. Conf. Parallel Processing, Aug. 1980, pp. 75-78.

[15] H. S. Stone, "Parallel processing and the perfect shuffle," IEEE Trans. Comp., vol. C-20, Feb. 1971, pp. 153-161.

MODELING OF SYNCHRONIZED ITERATIVE ALGORITHMS FOR MULTIPROCESSORS

FAYE A. BRIGGS and MICHEL DUBOIS
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

ABSTRACT

We develop an abstract model for a numerical computation and its decomposition into parallel tasks. The theoretical model is then used to evaluate the performance of iterative algorithms on an MIMD processor architecture. Analytical models are derived for communication through a time-shared bus or the shared memory. We focus on the determination of the optimum number of processors to be allocated to an iterative algorithm, using each of the two communication methods.

1. INTRODUCTION

Algorithms implemented on MIMD machines ("MIMD algorithms") are more difficult to study than those on SIMD machines. The difficulty stems from the increased flexibility. In a synchronized MIMD computation, processes operate in a speed-independent manner and have to wait on each other at interaction points. Probabilistic modeling techniques drawn from order statistics have to be used [WEI78, DAV70]. In some approaches, where the MIMD machine has been built, measurements are made on the actual computations, and an analytical model is derived to fit the performance of the actual machine [OLE78, BAU78]. While this method is valid for tuning existing systems, it is not so practical in theoretical comparisons of algorithms on various architectures. The difficulty is that the stochastic properties of the processing time of each process involved in an MIMD algorithm vary, not only with the problem itself, but also with the execution environment. In this paper we define an abstract model for an iterative numerical computation. This model distinguishes the two aforementioned influences on the processing time distribution and is a good approximation to the actual behavior of a process. This abstract model is then used to study iterative numerical algorithms on a multiprocessor system. The study is useful in comparing various implementations of an MIMD iterative algorithm and in finding the degree of decomposition optimum for a given implementation.

2. ITERATIVE NUMERICAL ALGORITHMS

Iterative numerical algorithms typify the MIMD algorithms studied in this paper. A general numerical iterative algorithm [DAH74] can be described by an iteration function [KUN76]:

$$x_{i+1} = \phi(x_i, x_{i-1}, \ldots, x_{i-d+1})$$

where, in general, $x \in R^n$.

The strategy chosen to implement this algorithm on an MIMD machine is to decompose it into P processes, each updating $\frac{n}{P}$ iterates. Each process is allocated to a different processor. If the computation of each processor for each iteration has the same time complexity, and if all the processors are identical, it is reasonable to assume that the processing times of each process have the same stochastic properties ("i.i.d. * tasks"). At the end of each iteration, the processors exchange some results and synchronize to initiate the next iteration. We will clarify this model in section 4.

3. SYSTEM ARCHITECTURE

Figure 1 displays the architecture modeled in this paper. This system consists of P processing elements (PEs) which communicate through a set of P shared memory modules and a high-speed bus. Each PE consists of a processor and a local memory with a DMA port to the bus or shared memory. The block transfer may take place concurrently with processing. A local memory contains code and data. Many MIMD computers have a structure similar to the one in Figure 1: the Siemens SMS [KOB76], the Papros [CON77], EMMA [STR79] and the MIMD mode of the PM4 [BRI79] are such examples. In the following section, a formal model is given for a numerical computation in such a system. C.mmp is different, mainly because of the interprocessor communication, which is implicit in the sharing of the main memory. The local memory is only used by the operating system [WUL72].

4. ABSTRACT MODEL FOR A HOMOGENEOUS NUMERICAL COMPUTATION

In the architecture described in the previous section, no memory conflict occurs within a PE, as the currently referenced code and data are in the private memory. The randomness in processing times is essentially due to the dependency on inputs and possible system interrupts (due to page faults, for example).

Let $\mathcal{A}$ be an algorithm to be decomposed and run on an MIMD machine of the type described above. In general, $\mathcal{A}$ contains a random number of atomic operations (this randomness occurs, for instance, when series are computed or when the evaluation of a function can be done in various ways, depending on the argument value). An atom is any basic arithmetic or logical operation.

When processed on a given machine, an atom of computation has a random duration because of variable cycle times for each basic operation, of operand-dependent operations (e.g. an addition may be aborted if one of the operands is null) and of system interrupts (for page fault, for example).

If the set of atoms in the algorithm $\mathcal{A}$ can be partitioned in PMAX independent subsets (called cells) with the same stochastic properties ("i.i.d." cells), then $\mathcal{A}$ is said to be homogeneous with respect to the cells.

To obtain a decomposition of a homogeneous computation in i.i.d. tasks, we divide the computation into groups containing the same number of cells. If P tasks are so created, we must have $P \leq PMAX$, and P is a divisor of PMAX. While the set of possible degrees of decomposition is discrete, it is usual to draw speed-up curves with P taken as a continuous variable. One must keep in mind that P can only assume values on a restricted set of integers.

For example, in the case of the Jacobi iterative algorithm for the solution of a large sparse system, a cell could be the computation of one iterate. The total number of atoms in a cell is fixed and constant (proportional to the dimension of the system); the time spent in interrupts

---

(*) For short, we denote the cumulative distribution function by c.d.f. and the density function by p.d.f.; i.i.d. stands for "independent and identically distributed."

during an atom is practically null if the only interrupts are for page faults (the code for a sum of products can be easily made very compact if a "do loop" is used). Finally, the distribution of the time taken by the basic operations will depend on the sparsity of the matrix. Figure 2 shows the individual behavior of a process involved in an MIMD algorithm.

5. MODELING OF THE PROCESSING TIMES

Given the abstract model of the previous section, a processing time distribution can be deducted for all $P \leq PMAX$, under certain conditions. Let m and σ^2 be the mean and the variance of the time taken by an atomic operation. They are both independent of P, since the processors do not interact during the processing. This would not be the case for C.mmp, where memory access conflicts influence the duration of the execution of an atom. Let M and Σ^2 be the mean and variance of N, the total number of atoms in a cell. To start with, we choose the case for which P = PMAX. A task is then a cell.

If the distributions of N and of the atom are independent (we assume independence throughout this paper for consistency and tractability), it can be shown (see [PAP65]) that

$$\overline{T}_{PMAX} = M\, m, \text{ and}$$

$$\sigma^2_{PMAX} = m^2\, \Sigma^2 + \sigma^2\, M \qquad (5.1)$$

where $\overline{T}_P$ and σ^2_P are the mean and variance of the processing time for P processors.

Let $k_P = \frac{PMAX}{P}$ be the number of cells in a task when the degree of decomposition is P. Given the homogeneous property of the computation, we have

$$\overline{T}_P = k_P\, M\, m = k_P\, \overline{T}_{PMAX} \qquad (5.2)$$

$$\sigma^2_P = m^2\, \Sigma^2\, k_P + \sigma^2\, k_P\, M = k_P\, \sigma^2_{PMAX} \qquad (5.3)$$

The coefficient of variation C_P of the processing time of each task is

$$C_P = \frac{\sigma_P}{\overline{T}_P} = \sqrt{\frac{P}{PMAX}}\, C_{PMAX} \qquad (5.4)$$

More can be said about the distribution of the processing time if N is constant and large enough. Indeed, by the central-limit theorem [PAP65], the p.d.f. of the processing time is then normal, with mean N m and variance N σ^2. This property can even be extended to the case where the atoms do not have the same distribution or when N is random [DUB80]. However, in this paper, we limit ourselves to the case when N is constant and assume that it is large enough for the central-limit theorem to be applicable. Figure 3 shows some processing time distributions for various values of the parameters.

6. MODELING OF COMMUNICATION BETWEEN PROCESSORS

At the end of each iteration, the processors have to communicate their results. Let $\mathcal{D}$ be the set of variables updated by all the processors for each iteration. We assume total symmetry and homogeneity. At the beginning of an iteration, each processor must have the most recent values of the variables in $\mathcal{D}$. We assume that $\mathcal{D}$ is independent of the degree of decomposition and that each processor produces the same amount of data for a given decomposition. We model the time of a transfer by a constant:

$$B_P = t_{CO} + \frac{t_{CD}}{P} \tag{6.1}$$

t_{CO} is the average overhead per transfer. It includes the time to set up a connection, and the time taken by some overhead processing (an eventual operating system call should be included in the processing time). t_{CD} is the time taken by the transfer of all the data in $\mathcal{D}$ (not counting overhead). For example, in the Jacobi Algorithm for solving a linear system of n equations, each block transfer involves $\frac{n}{P}$ data elements. t_{CD} is the time to store or fetch the n data as block to or from the shared memory or to put them on the bus.

6.1 Communication through the Shared Memory

A memory module in shared memory is associated with each of the P processors. At the completion of the computation for the current iteration, the process is blocked, and the processor activates a transfer to shared memory. When access to a module is granted, a connection path is established between the DMA gate and the memory module.

The results of the iteration are then transferred from the local memory to the shared module. When this transfer is completed, a signal is sent to all the other PEs. Each of these PEs can access the data when granted permission. In this scheme, a PE can read data from an already completed process's module while its processor is still computing.

A lower bound on I_P, the iteration time for P processors, is obtained as follows. When the Pth processor is done, it has to write its data in its associated module in the shared memory; only then can the (P-1) other processors read it in turn. So,

$$I_P \geq T_{P,P} + P\ B_P \tag{6.2}$$

where $T_{j,P}$ is the processing time of the jth processor to terminate (in the chronological order) when P processors are used.

There exists a schedule such that the bound (6.2) is actually achieved. In this schedule, all the processors wait for the last processor to terminate, their data are written in parallel in the P modules and then they read each other's data in an orderly fashion. This can be done in a time $P\ B_P$. So,

$$I_P = T_{P,P} + P\ B_P \tag{6.3}$$

If the Pth mean order statistic of $T_{P,P}$ can be obtained, we can write the mean iteration time, $\overline{I}_P$, as

$$\overline{I}_P = \overline{T}_{P,P} + P\ B_P \tag{6.4}$$

where the $\overline{T}_{P,P}$ is the mean of $T_{P,P}$.

6.2 Communication through the Time-shared Bus

When a processor has completed an iteration, it requests a bus transfer and goes into an idle state. When the bus is free, a connection is established on the bus to all other PEs, in time t_{CO}. The data from the requesting processor is broadcasted to all other PEs, and the transfer time is $\frac{t_{CD}}{P}$. In this case, when the jth processor (in chronological order) has completed its iteration, there remains at least (P-j+1) transfers to take place on the bus. As the first processor to finish always finds the bus free, we have

$$I_P = \max_{j=1,\dots,P} [T_{j,P} + (P-j+1)\, B_P].$$

Taking the mean of both sides, we obtain

$$\overline{I}_P = \overline{\max_{j=1,\dots,P}} [T_{j,P} + (P-j+1)\, B_P]$$

and

$$\overline{I}_P \simeq \max_{j=1,\dots,P} [\overline{T}_{j,P} + (P-j+1)\, B_P] \qquad (6.5)$$

Equation (6.5) is a lower bound on the iteration time and so gives an upper bound on the speed-up. However, simulations have shown that (6.5) is a very tight bound.

7. PERFORMANCE OF SYNCHRONIZED ITERATIVE ALGORITHMS

The mean order statistics of the processing times for the distributions considered in this paper can be put into the form

$$\overline{T}_{j,P} = \overline{T}_P + O_{j,P}\, \sigma_P, \text{ where} \qquad (7.1)$$

$\overline{T}_{j,P}$ is the mean jth order statistic of the processing time with P processors, $\overline{T}_P(\sigma_P^2)$ is the mean (variance) of the processing time with P processors, and $O_{j,P}$ is the jth mean order statistic among P for the distribution of the processing time with mean 0 and variance 1 [DAV70].

For a uniform distribution,

$$O_{j,P} = \sqrt{3}\,\frac{2j-P-1}{P+1} \qquad (7.2)$$

For an offset exponential distribution (an exponential plus a constant),

$$O_{j,P} = H_P - H_{P-j} - 1 \qquad (7.3)$$

where $H_i = \sum_{k=1}^{i} \frac{1}{k}$ is the ith harmonic number ([KNU73]).

For a normal $N(\overline{T}_P, \sigma_P)$, no analytical form has yet been found. $O_{j,P}$ can be tabulated [HAR61] or approximated by asymptotic results [DAV70]: if $\frac{j}{P} \simeq \frac{1}{2}$,

$$\lim_{P\to\infty} O_{j,P} = F^{-1}\left(\frac{j}{P+1}\right), \qquad (7.4)$$

where F(x) is the c.d.f. of the normal N (0,1).

In the case of the normal distribution, negative processing times can occur. This problem could be solved by truncating the p.d.f. at 0, at the price of further complications for computing the mean order statistics. We have chosen to use a normal approximation and to neglect the effect of negative processing times if $C_{PMAX}<.5$.

Applying (5.2) and (5.3) to (7.1), we obtain

$$\overline{T}_{j,P} = \frac{PMAX}{P}\,\overline{T}_{PMAX} + \sqrt{\frac{PMAX}{P}}\,\sigma_{PMAX}\,O_{j,P} \qquad (7.5)$$

where we recall that PMAX, $\overline{T}_{PMAX}$ and σ_{PMAX} are the maximum degree of decomposition and the mean and standard deviation of the processing time of a cell.

As the overall computation is the sum of the PMAX cells, the mean running time of the iterative algorithm on a uniprocessor is

$$\overline{I}_1 = \overline{T}_1 = PMAX\,\overline{T}_{PMAX} \qquad (7.6)$$

and (7.5) becomes

$$\overline{T}_{j,P} = \frac{\overline{T}_1}{P}\left[1 + \sqrt{\frac{P}{PMAX}}\,C_{PMAX}\,O_{j,P}\right] \qquad (7.7)$$

Combining (7.7), (6.4) and (6.5), we obtain the following results.

7.1 Speed-up for Communication through Shared Memory

The mean iteration time when communication is through the shared memory is

$$\overline{I}_P = \frac{\overline{T}_1}{P}\left[1 + \sqrt{\frac{P}{PMAX}}\,C_{PMAX}\,O_{P,P} + P^2\,t'_{CO} + P\,t'_{CD}\right] \qquad (7.8)$$

where $t'_{CO} = \frac{t_{CO}}{\overline{T}_1}$ and $t'_{CD} = \frac{t_{CD}}{\overline{T}_1}$ are the normalized overhead time and the normalized data transfer time (t'_{CO} is very small in practice, since t_{CO} is the overhead per block transfer; t'_{CD} is the normalized time required to move all the data, excluding overhead).

The average speed-up is

$$S_P^{MEM} = \frac{P}{1 + \sqrt{\frac{P}{PMAX}}\,C_{PMAX}\,O_{P,P} + P^2\,t'_{CO} + P\,t'_{CD}} \qquad (7.9)$$

The second, third and fourth terms of the denominator represent the penalty incurred by synchronization, transfer overhead and transfer of the data, respectively.

7.2. Speed-up for Communication on the Bus

Similarly, we have

$$\overline{T}_P \approx \frac{\overline{T}_1}{P} \max_{j=1,\ldots,P} [1 + \sqrt{\frac{P}{PMAX}}\, C_{PMAX}\, O_{j,P} + (P-j+1)\,(P\, t'_{CO} + t'_{CD})]$$

and,

$$S_P^{BUS} \approx \frac{P}{\max\limits_{j=1,\ldots,P} [1 + \sqrt{\frac{P}{PMAX}}\, C_{PMAX}\, O_{j,P} + (P-j+1)\,(P\, t'_{CO} + t'_{CD})]} \qquad (7.10)$$

The speed-up of formula (7.9) and (7.10) are plotted respectively in Figures 4 and 5. The distributions for the processing time are normal, uniform and offset exponential. For the normal distribution, the curves obtained with the asymptotic values for $O_{j,P}$ are compared to the one obtained with the tabulated values [HAR61].

8. CONCLUSION

MIMD architectures are well-suited to the parallel processing of synchronized iterative algorithms. For n iterates and P processors, the computation of $\frac{n}{P}$ iterates is allocated to each of the P processors. Processors can communicate through a set of P shared memory modules or a high-speed bus in block transfer mode. A model for the atomic behavior of a process in a numerical computation has led us to consider the normal distribution as an asymptotic approximation for the processing time distribution. The additional hypothesis of i.i.d. tasks for all degrees of decomposition also shows that the coefficient of variation of the processing time varies as the square root of P. The results for uniform and exponential distributions have been compared to the ones for the normal distribution.

The uniform distribution approximation is quite good and does not require any tabulation. Another method is to apply asymptotic results of order statistics, which gives very good approximations but does not yield an analytical result. The curves obtained with an offset exponential distribution are less reliable and underestimate the speed-up. Figures 4 and 5 also show the influence of the overhead term t'_{CO} on the speed-up curves, and compare communication through the bus and through shared memory. For the same values of the parameters t'_{CO} and t'_{CD}, the bus system outperforms the communication through memory. This is due to the fact that P^2 transfers take place in the second case, while only P is necessary for the first case, due to the broadcasting capability of the bus. However, the overhead term t_{CO} might be larger in the case of the bus. These observations justify, for example, the special care given to data exchanges in the SMS computer [KOB76]. The speed-up curves of Figures 4 and 5 can be used as a guideline by the operating system or the user for an effective decomposition of an MIMD iterative problem into tasks of similar statistical properties. It is more realistic than usual complexity studies, which do not include overhead, and it can serve to compare theoretically the performance of various architectures for the implementation of iterative algorithms.

REFERENCES

[BAU78] G.M. BAUDET, "The Design and Analysis of Algorithms for Asynchronous Multiprocessors," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA., Apr. 1978.

[BRI79] F.A. BRIGGS, K.S. FU, K. HWANG, and J.H. PATEL, "PM4 - A Reconfigurable Multiprocessor System for Pattern Recognition and Image Processing," Proceeding of the AFIPS, Vol. 48, pp. 255-265.

[CON77] V. CONRAD and Y. WALLACH, "Iterative Solution of Linear Equations on a Parallel Processor System," IEEE Transactions on Computers, Vol. C-25, Sept. 1977, pp. 838-847.

[DAH] G. DAHLQUIST and A. BJORCK, Numerical Methods. Englewood Cliffs, N.J. : Prentice-Hall inc., 1974.

[DAV70] H.A. DAVID, Order Statistics. New York: John Wiley, 1970.

[DUB80] M. DUBOIS and F.A. BRIGGS, "Performance of Synchronized Iterative Algorithms in Multiprocessor Systems," Purdue University, Technical Report (to be published).

[HAB76] A.N. HABERMANN, Introduction to Operating System Design. Chicago: Science Research Associates Inc., 1976.

[HAR61] H. L. HARTER, "Expected Values of Normal Order Statistics," Biometrika, 48, 1 and 2, 1961, pp. 151, 165.

[KOB76] R. KOBER, "A Fast Communication Processor for the SMS Multimicroprocessor System," Euromicro, 1976, North-Holland Publishing Company.

[KNU73] D.E. KNUTH, The Art of Computer Programing - Vol. 1: Fundamental Algorithms. Addison Wesley, 1973.

[KUN76] H.T. KUNG, "Synchronized and Asynchronous Parallel Algorithms for Multiprocessors," in Algorithms and Complexity: New Directions and Recent Results. J.F. Traub Ed., New York: Academic Press, 1976.

[OLE79] P.N. OLEINICK, "The Implementation and Evaluation of Parallel Algorithms on C.mmp," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1979

[PAT76] J.K. PATEL, C.H. KAPADIA, and D.B. OWEN, Handbook of Statistical Distributions. New York: Marcel Dekker Inc., 1976.

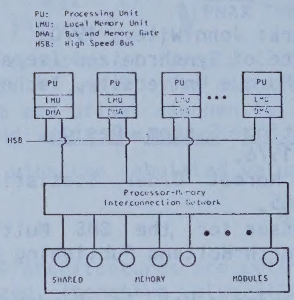
[PAP65] A. PAPOULIS, Probability, Random Variables, and Stochastic Processes. McGraw-Hill, 1965.

[ROB79] J.T. ROBINSON "Some Analysis Techniques for Asynchronous Multiprocessor Algorithms," IEEE Transactions on Software Engineering, Vol. SE-5, Jan. 1979, pp. 24-30.

[STR79] L. STRINGA, "EMMA: A Multiprocessor on the Field," internal report, Electronica Son Giorgio-Elsag. S.p.A, Geneva.

[WEI788] B.W. WEIDE, "Statistical Methods in Algorithm Design and Analysis," Ph.D. Thesis, Department of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, Aug., 1978.

[WUL72] W.A. WULF and C.G. BELL, "C.mmp - A Multiminiprocessor," Proceeding of the AFIPS, Vol. 41, pp. 765-777.


PU: Processing Unit
LMU: Local Memory Unit
DMA: Bus and Memory Gate
HSB: High Speed Bus
PU
LMU
DMA
HSB
Processor-Memory Interconnection Network
SHARED
MEMORY
MODULES

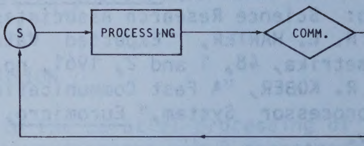

S
PROCESSING
COMM.

Figure 1

Multiprocessor System Architecture

Figure 2

Behavior of one processor in an MIMD algorithm. The synchronization is implied in communication

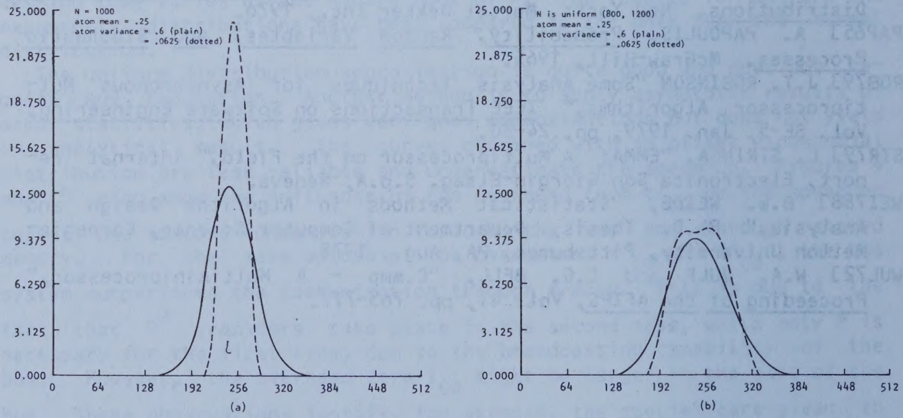


Figure 3

Processing time histograms for cells with various values of the parameters

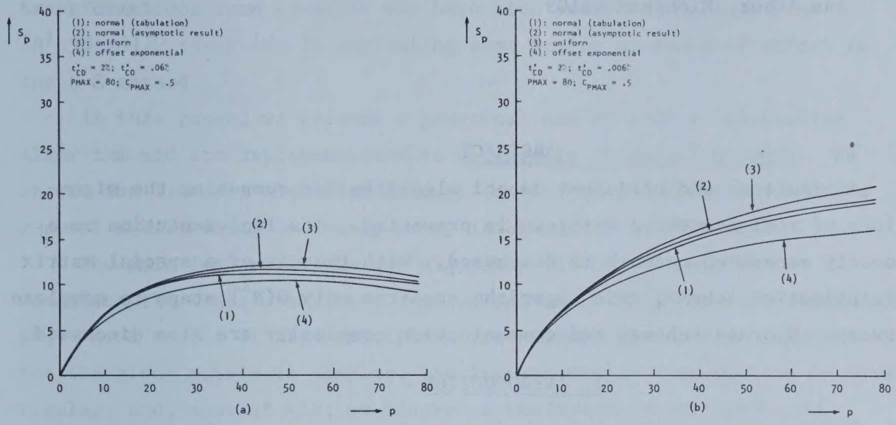


Figure 4

Speed-up versus decomposition for communication through the shared memory

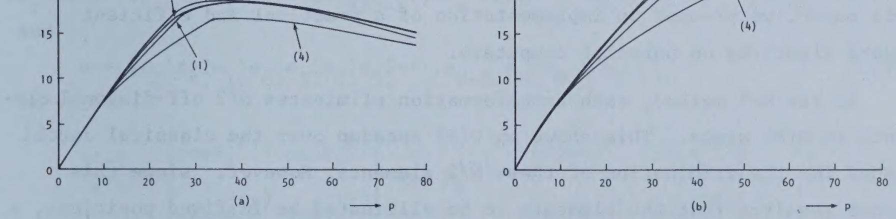


Figure 5

Speed-up versus decomposition for communication through the bus

A JACOBI ALGORITHM AND ITS IMPLEMENTATION ON PARALLEL COMPUTERS

KUO-WEI CHEN and KEKI B. IRANI
The University of Michigan
Ann Arbor, Michigan 48109

ABSTRACT

A practical and efficient Jacobi algorithm for computing the eigenvalues of real symmetric matrices is presented. Its implementation on a linearly connected network is discussed. With the use of a special matrix multiplication scheme, this algorithm requires only $O(N^2)$ steps to complete a sweep. Storage schemes and communication complexity are also discussed.

1. INTRODUCTION

In the classical Jacobi method, a real symmetric matrix is reduced to the diagonal form by a sequence of elementary orthogonal transformations [7]. Let the matrix be of order N, then each transformation requires O(N) steps. A sequence of N(N-1)/2 transformations which eliminates every one of the N(N-1)/2 off-diagonal elements in the upper triangle exactly once, is called a sweep. The computational complexity of a sweep is thus $O(N^3)$. A parallel implementation of a Jacobi algorithm on Illiac IV machines is discussed by Kuck and Sameh [2], which will be referred to in this paper as the K-S method. Sameh [6] shows an algorithm which is optimal in the sense that minimum number of transformations are used to complete a sweep. In this paper, we present an implementation of a practical and efficient Jacobi algorithm on parallel computers.

In the K-S method, each transformation eliminates N/2 off-diagonal elements in O(N) steps. This shows an O(N) speedup over the classical Jacobi method for the elimination of these N/2 elements. However, since this method requires that the elements to be eliminated be in fixed positions, a permutation of the matrix is necessary after each transformation in order to bring other elements to these positions for annihilation. It can be shown that, using the permutations of the K-S method, $(N-1)^2$ transformations and $(N-1)^2$ permutations, and thus $O(N^3)$ steps, are needed to ensure that every element has been annihilated at least once. It has also been indicated in [5] that usually 6 to 10 sweeps are needed to diagonalize a real symmetric matrix (also see [3]). Hence the overall performance of the K-S method does not show much improvement over the serial Jacobi method. Moreover, due to the nature of this permutation, in $(N-1)^2$ transformations some elements may have been annihilated as many as $(N^2-5N+10)/2$ times $(N>2)$, indicating some degree of waste of effort in the K-S method.

In this paper, we present a practical and efficient Jacobi-like algorithm and its implementation on a linearly connected network. We choose the linearly connected network for its simple structure and its general availability in nearly every interconnection network.

The main feature of the new algorithm is the scheme used for computing the transformations, namely, matrix multiplication by diagonals [4]. As a result of this multiplication scheme, the storage scheme for the given matrix is simpler, the interprocessor communiction is more regular, and, most of all, no longer is the broadcasting capability required from the interconnection network. Also, to complete a sweep, only N transformations and N/2 permutations are needed; so the complexity is $O(N^2)$ per sweep.

2. A SCHEMATIC OF THE NEW METHOD

Let A be a real symmetric matrix of even order N. The new method reduces A into diagonal form through a sequence of orthagonal transformations, $A_{r+1} = \phi_r A_r \phi_r^t$ $(r = 0, 2, 4, \dots)$ and $A_{r+1} = \rho_r A_r \rho_r^t$ $(r=1, 3, 5, \dots)$, and permutations $\sigma A_{r+1} \sigma^t$ $(r=1, 3, 5, \dots)$ where $A_0 = A$ and

$$\phi_r = \text{diag.}\ (T_0^{(r)}, T_1^{(r)}, \dots, T_{N/2-1}^{(r)}),\quad \rho_r = \text{diag.}\ (1, R_1^{(r)}, R_2^{(r)}, \dots, R_{N/2-1}^{(r)}, 1) \qquad (1)$$

and†.

$$\sigma = (e_2 \vdots e_4 \vdots e_1 \vdots e_6 \vdots e_3 \vdots e_8 \vdots e_5 \vdots \cdots \vdots e_{N-5} \vdots e_N \vdots e_{N-3} \vdots e_{N-1})^t \qquad (2)$$

in which

$$T_k^{(r)} = \begin{pmatrix} \cos\theta_k^{(r)} & \sin\theta_k^{(r)} \\ -\sin\theta_k^{(r)} & \cos\theta_k^{(r)} \end{pmatrix} \text{ and } R_k^{(r)} = \begin{pmatrix} \cos\alpha_k^{(r)} & \sin\alpha_k^{(r)} \\ -\sin\alpha_k^{(r)} & \cos\alpha_k^{(r)} \end{pmatrix}, \qquad (3)$$

and $\theta_k^{(r)}$ and $\alpha_k^{(r)}$ are determined respectively from the diagonal submatrics of A_r,

$$\begin{bmatrix} a_{2k,2k}^{(r)} & a_{2k,2k+1}^{(r)} \\ a_{2k,2k+1}^{(r)} & a_{2k+1,2k+1}^{(r)} \end{bmatrix} \ (0 \le k \le N/2-1) \text{ and } \begin{bmatrix} a_{2k-1,2k-1}^{(r)} & a_{2k-1,2k}^{(r)} \\ a_{2k-1,2k}^{(r)} & a_{2k,2k}^{(r)} \end{bmatrix} (1 \le k \le N/2-1)$$

---

†: e_i is the i^{th} unit column vector.

such that

$$\tan 2\theta_k^{(r)} = 2a_{2k,2k+1}^{(r)} \Big/ \left[a_{2k,2k}^{(r)} - a_{2k+1,2k+1}^{(r)}\right] \text{ and } \left|\theta_k^{(r)}\right| \le \pi/4,$$

$$\text{and } \tan 2\alpha_k^{(r)} = 2a_{2k-1,2k}^{(r)} \Big/ \left[a_{2k-1,2k-1}^{(r)} - a_{2k,2k}^{(r)}\right] \text{ and } \left|\alpha_k^{(r)}\right| \le \pi/4. \qquad (4)$$

The transformation $\phi_r A_r \phi_r^t$, which is the same as in the K-S method, eliminates N/2 elements of A_r in positions $(2k,2k+1)$, $0 \le k \le N/2-1$. $\rho_r A_r \rho_r^t$, a new transformation, further eliminates N/2-1 elements of A_r in positions $(2k-1,2k)$, $1 \le k \le N/2-1$. A schematic of this method now follows.

```
step 0, $H \leftarrow A$;
step 1, compute $\phi$ from H;
step 2, compute $H_1 \leftarrow \phi H \phi^t$; /* eliminates N/2 elements in $(2k,2k+1), 0 \le k \le N/2-1$*/
step 3, compute $\rho$ from $H_1$;
step 4, compute $H_2 \leftarrow \rho H_1 \rho^t$; /* eliminates N/2-1 elements in $(2k-1,2k)$,
        $1 \le k \le N/2-1$*/
step 5, Perform $H \leftarrow \sigma H_2 \sigma^t$; /* brings N-1 new elements to super diagonal */
step 6, if A has converged, terminate;
        otherwise go to step 1;
```

Thus, at the completion of step 4, each of the N-1 elements of matrix H in superdiagonal positions $(k,k+1)$, $0 \le k \le N-2$, has been eliminated exactly once. The permutation $\sigma A_r \sigma^t$ in step 5 has an interesting property that, after each permutation, N-1 new off-diagonal elements emerge in the superdiagonal positions for annihilation and, after N/2 permutations, each of N(N-1)/2 off-diagonal elements, having been annihilated exactly once, returns to its original position. Hence only N transformations (or, N/2 pair of transformations) and N/2 permutations are needed to complete a "sweep".

3. COMPUTING THE TRANSFORMATION BY DIAGONALS

The partitioning scheme used in the K-S method for computing the transformation $\phi_r A_r \phi_r^t$ requires that the elements of the matrix ϕ be broadcast to all processing elements that need them. In the new algorithm, the matrix multiplication is performed by successively computing the diagonals of the product matrix directly from the elements of matrices A_r and ϕ_r. In this way not only no broadcasting capability is required of the interconnection network, but also the storage scheme for A is greatly simplified, as will be shown below.

3.1 Computing the Transformation $\phi_r A_r \phi_r^t$

For notational convenience, we shall, in what follows, drop the subscripts r and the superscripts (r) and rewrite the transformation as $B=\phi A\phi^t$, where ϕ is defined by (1), (3) and (4). Also, to describe the parallel features of the algorithm, we append an inequality, of the form $(x\le i\le y)$, to a statement which is to be executed simultaneously for all values of the index in the specified range. Furthermore, 0-origin is used in indexing the elements of a matrix.

The derivation of the expressions for the diagonals $b_{i,i+k}$ $(o\le k\le N-1)$ is somewhat tedious to describe mathematically. Rather, we give only some useful information in Appendix . The algorithm for computing the transformation $B=\phi A\phi^t$ now follows.

Algorithm for $B=\phi A\phi^t$

```
/* First determine the angles of rotations */
 i. Solve $\tan 2\theta_{\frac{i-1}{2}} = 2a_{i-1,i}/(a_{i-1,i-1}-a_{ii})$ and $|\theta_{\frac{i-1}{2}}|\le \pi/4$ $(0\le i\le N-1, i:\text{odd})$
 ii. Compute $\cos\theta_{\frac{i-1}{2}}$ and $\sin\theta_{\frac{i-1}{2}}$ $(0\le i\le N-1, i:\text{odd})$

/* The main diagonal */
iii. $b_{ii}=\begin{cases} a_{ii}\cos^2\theta_{\frac{i}{2}} +2a_{i,i+1}\cos\theta_{\frac{i}{2}}\sin\theta_{\frac{i}{2}} +a_{i+1,i+1}\sin^2\theta_{\frac{i}{2}} & (0\le i\le N-1, i:\text{even}) \\ a_{ii}\cos^2\theta_{\frac{i-1}{2}} -2a_{i-1,i}\cos\theta_{\frac{i-1}{2}}\sin\theta_{\frac{i-1}{2}} +a_{i-1,i-1}\sin^2\theta_{\frac{i-1}{2}} & (0\le i\le N-1, i:\text{odd}) \end{cases}$

/* The superdiagonal. $b_{2k,2k+1}$ are set to 0 as a result of annihilation */
iv. $b_{i,i+1}=\begin{cases} 0 & (0\le i\le N-2, i:\text{even}) \\ -\sin\theta_{\frac{i-1}{2}}(a_{i-1,i+1}\cos\theta_{\frac{i+1}{2}} + a_{i-1,i+2}\sin\theta_{\frac{i+1}{2}}) \\ \quad +\cos\theta_{\frac{i-1}{2}}(a_{i,i+1}\cos\theta_{\frac{i+1}{2}} + a_{i,i+2}\sin\theta_{\frac{i+1}{2}}) & (0\le i\le N-2, i:\text{odd}) \end{cases}$

/* Other diagonals */
 for k=2 step 1 until N-1 do begin
     if k:even then
v.       $b_{i,i+k}=\begin{cases} \cos\theta_{\frac{i}{2}}(a_{i,i+k}\cos\theta_{\frac{i+k}{2}} + a_{i,i+k+1}\sin\theta_{\frac{i+k}{2}}) \\ \quad +\sin\theta_{\frac{i}{2}}(a_{i+1,i+k}\cos\theta_{\frac{i+k}{2}} + a_{i+1,i+k+1}\sin\theta_{\frac{i+k}{2}}) & (0\le i\le N-k-1, i:\text{even}) \\ \sin\theta_{\frac{i-1}{2}}(a_{i-1,i+k-1}\sin\theta_{\frac{i+k-1}{2}} - a_{i-1,i+k}\cos\theta_{\frac{i+k-1}{2}}) \\ \quad -\cos\theta_{\frac{i-1}{2}}(a_{i,i+k-1}\sin\theta_{\frac{i+k-1}{2}} - a_{i,i+k}\cos\theta_{\frac{i+k-1}{2}}) & (0\le i\le N-k-1, i:\text{odd}) \end{cases}$
     else    /* k :odd */
vi.      $b_{i,i+k}=\begin{cases} -\cos\theta_{\frac{i}{2}}(a_{i,i+k-1}\sin\theta_{\frac{i+k-1}{2}} - a_{i,i+k}\cos\theta_{\frac{i+k-1}{2}}) \\ \quad -\sin\theta_{\frac{i}{2}}(a_{i+1,i+k-1}\sin\theta_{\frac{i+k-1}{2}} - a_{i+1,i+k}\cos\theta_{\frac{i+k-1}{2}}) & (0\le i\le N-k-1, i:\text{even}) \\ -\sin\theta_{\frac{i-1}{2}}(a_{i-1,i+k}\cos\theta_{\frac{i+k}{2}} + a_{i-1,i+k+1}\sin\theta_{\frac{i+k}{2}}) \\ \quad +\cos\theta_{\frac{i-1}{2}}(a_{i,i+k}\cos\theta_{\frac{i+k}{2}} + a_{i,i+k+1}\sin\theta_{\frac{i+k}{2}}) & (0\le i\le N-k-1, i:\text{odd}) \end{cases}$
```

```
end;
```

End of Algorithm for $B=\phi A\phi^t$

The computational complexity of the above algorithm on a linearly connected network of N processing elements is readily seen to be O(N).

3.1.1 Storage Scheme For Matrix A. In parallel processing, the storage scheme for the data of the algorithm affects the cost of interprocessor communication [1]. A good storage scheme may reduce or even minimize this communication cost, while a bad one may significantly degrade the performance of the algorithm on the given parallel computer. For the computation of the transformation $B=\phi A\phi^t$, the algorithm given in Section 3.1 clearly suggests that matrix A be stored by columns, i.e. column i of A be stroed in the memory of the i^{th} processing element. The complete storage scheme for matrix A is as follows:

i. $a_{ii} \rightarrow$ loc. 0, PE_i $(0 \le i \le N-1)$

ii. $a_{i,i+1} \rightarrow$ loc. 1, PE_{i+1} $(0 \le i \le N-2$, i:even)

iii. for all k, $2 \le k \le N-2$ and k:even

$a_{i,i+k} \rightarrow$ loc. k, PE_{i+k} $(0 \le i \le N-k-1$, i:even)

$a_{i,i+k} \rightarrow$ loc. k+1, PE_{i+k} $(0 \le i \le N-k-1$, i:odd)

iv. for all k, $3 \le k \le N-1$ and k:odd

$a_{i,i+k} \rightarrow$ loc. k-1, PE_{i+k} $(0 \le i \le N-k-1$, i:even)

v. for all k, $1 \le k \le N-3$ and k:odd

$a_{i,i+k} \rightarrow$ loc. k+2, PE_{i+k} $(0 \le i \le N-k-1$, i:odd)

An example of this storage scheme for a 16x16 matrix A is given in Figure 1.

3.1.2 Communication Complexity. As a result of the above storage scheme, the analysis of the interprocessor communication is simple and straightforward. All that is needed to determine the data transfer required for aligning the operands of a statement is to examine and compare the column indices of the operands. For example, to execute the first step of the algorithm, all $a_{i-1,i-1}$ $(0 \le i \le N-1$, i:odd) have to be moved to PE_i from PE_{i-1}. The communication requirement for the computation of $B=\phi A\phi^t$ follows.

```
/* Assume the angles of rotations are to be computed in $PE_i$, $0 \le i \le N-1$, i :odd */
i. $a_{i-1,i-1}$ : $PE_{i-1}$ ---> $PE_i$   ($0 \le i \le N-1$, i :odd)
/* After $\cos\theta_{\frac{i-1}{2}}$ and $\sin\theta_{\frac{i-1}{2}}$ ($0 \le i \le N-1$, i :odd) are computed in the above $PE_i$,
   a copy of each is moved to $PE_{i-1}$ with two transfers. These four-element
   groups then form the 2x2 submatrices $T_k$, $0 \le k \le N/2-1$, of $\phi_r$.        */
ii. $\cos\theta_{\frac{i-1}{2}}$ , $\sin\theta_{\frac{i-1}{2}}$ : $PE_i$ ---> $PE_{i-1}$   ($0 \le i \le N-1$, i :odd)
/* We show one way of computing this statement */
iii. $a_{i,i+1}$ ,$a_{i+1,i+1}$ : $PE_{i+1}$ ---> $PE_i$   ($0 \le i \le N-1$, i :even)
     $a_{i-1,i-1}$             : $PE_{i-1}$ ---> $PE_i$   ($0 \le i \le N-1$, i :odd)
/* $\cos\theta_{\frac{i+1}{2}}$ is in both $PE_{i+1}$ and $PE_{i+2}$, but $\cos\theta_{\frac{i-1}{2}}$ is in both $PE_{i-1}$ and $PE_i$ */
iv. $\cos\theta_{\frac{i-1}{2}}$, $\sin\theta_{\frac{i-1}{2}}$ : $PE_i$ ---> $PE_{i+1}$   ($0 \le i \le N-2$, i :odd) /* must be saved in $PE_{i+1}$ */
    $a_{i-1,i+2}$, $a_{i,i+2}$ : $PE_{i+2}$ ---> $PE_{i+1}$   ($0 \le i \le N-2$, i :odd)
    for k = 2 step 1 until N-1 do begin
        if k :even then
/* need data transfer only for i :odd and k :even */
/* $\cos\theta_{\frac{i-1}{2}}$ and $\sin\theta_{\frac{i-1}{2}}$ have been saved in $PE_{i+k-1}$ in the previous transfer */
v. $\cos\theta_{\frac{i-1}{2}}$,$\sin\theta_{\frac{i-1}{2}}$ ,$a_{i-1,i+k-1}$,$a_{i,i+k-1}$ : $PE_{i+k-1}$ ---> $PE_{i+k}$   ($0 \le i \le N-k-1$, i :odd)
                                                              /* All values must be saved */
        else    /* for i :odd and k :odd */
vi. $\cos\theta_{\frac{i-1}{2}}$ ,$\sin\theta_{\frac{i-1}{2}}$ : $PE_{i+k-1}$ ---> $PE_{i+k}$   ($0 \le i \le N-k-1$, i :odd) /* must be saved */
    $a_{i-1,i+k-1}$ ,$a_{i,i+k-1}$ : $PE_{i+k-1}$ ---> $PE_{i+k}$   ($0 \le i \le N-k-1$, i :odd)
    end;
```

PE

loc.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0 0	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8	9 9	10 10	11 11	12 12	13 13	14 14	15 15
1		0 1		2 3		4 5		6 7		8 9		10 11		12 13		14 15
2			0 2	0 3	2 4	2 5	4 6	4 7	6 8	6 9	8 10	8 11	10 12	10 13	12 14	12 15
3			1 2	1 3	3 4	3 5	5 6	5 7	7 8	7 9	9 10	9 11	11 12	11 13	13 14	13 15
4					0 4	0 5	2 6	2 7	4 8	4 9	6 10	6 11	8 12	8 13	10 14	10 15
5					1 4	1 5	3 6	3 7	5 8	5 9	7 10	7 11	9 12	9 13	11 14	11 15
6							0 6	0 7	2 8	2 9	4 10	4 11	6 12	6 13	8 14	8 15
7							1 6	1 7	3 8	3 9	5 10	5 11	7 12	7 13	9 14	9 15
8									0 8	0 9	2 10	2 11	4 12	4 13	6 14	6 15
9									1 8	1 9	3 10	3 11	5 12	5 13	7 14	7 15
10											0 10	0 11	2 12	2 13	4 14	4 15
11											1 10	1 11	3 12	3 13	5 14	5 15
12													0 12	0 13	2 14	2 15
13													1 12	1 13	3 14	3 15
14															0 14	0 15
15															1 14	1 15

Figure 1. Storage scheme for N= 16 (only indices are shown).

From the above analysis, it is easy to see that only two types of data transfers or routings are needed, namely, transfers with shift distance +1 and -1; and no more than four routings are needed for each computation step. (This is also a reason why a linearly connected network is well suited for this problem.) Also, $\cos\theta_{\frac{i-1}{2}}$ and $\sin\theta_{\frac{i-1}{2}}$ of ϕ are shifted down the processor array one PE at a time, rather than broadcast to all PEs at one time. The communication complexity is, clearly, $O(N)$.

3.2 Computing The Transformation $\rho_r A_r \rho_r^t$

Although the transformation matrix ρ_r has a structure very similar to ϕ_r, it would be rather complex to compute $\rho_r A_r \rho_r^t$ using the K-S method because of the partitioning technique used in the matrix multiplication and the stroage scheme used for matrix A. However, it makes almost no difference if we compute $\rho_r A_r \rho_r^t$ by diagonals.

We again rewrite the transformation as $B=\rho A\rho^t$ where ρ is defined by (1), (3) and (4). The main difference between the two algorithms is the computation of the first row, b_{0i}, and the last column $b_{i,N-1}$.

```
Algorithm for $B=\rho A\rho^t$

/* Determine the angles of rotations */
  i. Solve $\tan 2\alpha_{\frac{i}{2}} = 2a_{i-1,i}/(a_{i-1,i-1} - a_{ii})$ and $|\alpha_{\frac{i}{2}}| \le \pi/4$  $(1 \le i \le N-2, i\ :\text{even})$
  ii. Compute $\cos\alpha_{\frac{i}{2}}$ and $\sin\alpha_{\frac{i}{2}}$  $(1 \le i \le N-2, i\ :\text{even})$
/* First row except for $b_{00}=a_{00}$ and $b_{0,N-1}=a_{0,N-1}$ */
  iii. $b_{0i} = \begin{cases} a_{0i}\cos\alpha_{\frac{i+1}{2}} + a_{0,i+1}\sin\alpha_{\frac{i+1}{2}} & (1 \le i \le N-2, i\ :\text{odd}) \\ a_{0i}\cos\alpha_{\frac{i}{2}} - a_{0,i-1}\sin\alpha_{\frac{i}{2}} & (1 \le i \le N-2, i\ :\text{even}) \end{cases}$
/* Main diagonal except for $b_{N-1,N-1}=a_{N-1,N-1}$ */
  iv. $b_{ii} = \begin{cases} a_{ii}\cos^2\alpha_{\frac{i+1}{2}} + 2a_{i,i+1}\cos\alpha_{\frac{i+1}{2}}\sin\alpha_{\frac{i+1}{2}} + a_{i+1,i+1}\sin^2\alpha_{\frac{i+1}{2}} & (1 \le i \le N-2, i\ :\text{odd}) \\ a_{ii}\cos^2\alpha_{\frac{i}{2}} - 2a_{i-1,i}\cos\alpha_{\frac{i}{2}}\sin\alpha_{\frac{i}{2}} + a_{i-1,i-1}\sin^2\alpha_{\frac{i}{2}} & (1 \le i \le N-2, i\ :\text{even}) \end{cases}$
/* The superdiagonal; $b_{2k-1,2k}$ are set to 0 */
  v. $b_{i,i+1} = \begin{cases} 0 \quad (1 \le i \le N-2, i\ :\text{odd}) \\ \cos\alpha_{\frac{i+2}{2}}(-a_{i-1,i+1}\sin\alpha_{\frac{i}{2}} + a_{i,i+1}\cos\alpha_{\frac{i}{2}}) \\ \quad -\sin\alpha_{\frac{i+2}{2}}(a_{i-1,i+2}\sin\alpha_{\frac{i}{2}} - a_{i,i+2}\cos\alpha_{\frac{i}{2}}) & (1 \le i \le N-2, i\ :\text{even}) \\ -a_{N-3,N-1}\sin\alpha_{\frac{N-2}{2}} + a_{N-2,N-1}\cos\alpha_{\frac{N-2}{2}} & i = N-2 \end{cases}$
/* Other diagonals */
```

```
    for k = 2 step 1 until N-2 do begin
        if k :even then
vi.         $b_{i,i+k} = \begin{cases} \cos\alpha_{\frac{i+k+1}{2}} (a_{i+1,i+k} \sin\alpha_{\frac{i+1}{2}} + a_{i,i+k} \cos\alpha_{\frac{i+1}{2}}) \\ \quad + \sin\alpha_{\frac{i+k+1}{2}} (a_{i+1,i+k+1} \sin\alpha_{\frac{i+1}{2}} + a_{i,i+k+1} \cos\alpha_{\frac{i+1}{2}}) & (1 \le i < N-k-1, i : odd) \\ a_{N-k,N-1} \sin\alpha_{\frac{N-k}{2}} + a_{N-k-1,N-1} \cos\alpha_{\frac{N-k}{2}} & i = N-k-1 \\ \sin\alpha_{\frac{i+k}{2}} (a_{i-1,i+k-1} \sin\alpha_{\frac{i}{2}} - a_{i,i+k-1} \cos\alpha_{\frac{i}{2}}) \\ \quad + \cos\alpha_{\frac{i+k}{2}} (-a_{i-1,i+k} \sin\alpha_{\frac{i}{2}} + a_{i,i+k} \cos\alpha_{\frac{i}{2}}) & (1 \le i \le N-k-1, i : even) \end{cases}$
        else  /* k :odd */
vii.        $b_{i,i+k} = \begin{cases} -\sin\alpha_{\frac{i+k}{2}} (a_{i+1,i+k-1} \sin\alpha_{\frac{i+1}{2}} + a_{i,i+k-1} \cos\alpha_{\frac{i+1}{2}}) \\ \quad + \cos\alpha_{\frac{i+k}{2}} (a_{i+1,i+k} \sin\alpha_{\frac{i+1}{2}} + a_{i,i+k} \cos\alpha_{\frac{i+1}{2}}) & (1 \le i \le N-k-1, i : odd) \\ -\cos\alpha_{\frac{i+k+1}{2}} (a_{i-1,i+k} \sin\alpha_{\frac{i}{2}} + a_{i,i+k} \cos\alpha_{\frac{i}{2}}) \\ \quad + \sin\alpha_{\frac{i+k+1}{2}} (-a_{i-1,i+k-1} \sin\alpha_{\frac{i}{2}} + a_{i,i+k+1} \cos\alpha_{\frac{i}{2}}) & (1 \le i < N-k-1, i : even) \\ -a_{N-k-2,N-1} \sin\alpha_{\frac{N-k-1}{2}} + a_{N-k-1,N-1} \cos\alpha_{\frac{N-k-1}{2}} & i = N-k-1 \end{cases}$
    end;
End of Algorithm for $B=\sigma A\sigma^t$
```

The computational complexity of this algorithm is, again, O(N).

The analysis of the interprocessor communication is similar to that for computing $\phi A\phi^t$ and is thus omitted. The complexity is also O(N).

4. THE PERMUTATION $\sigma A_r\sigma^t$

We mentioned earlier that in the K-S method, in $(N-1)^2$ transformations some elements may have been annihilated as many as $(N^2-5N+10)/2$ times $(N>2)$. To overcome this inefficiency, we use a new permutation σ defined by (2), which is repeated below for convenience.

$$\sigma=(e_2\vdots e_4\vdots e_1\vdots e_6\vdots e_3\vdots e_8\vdots e_5\vdots \cdots \vdots e_{N-5}\vdots e_N\vdots e_{N-3}\vdots e_{N-1})^t \tag{2}$$

The graphical representation of σ for N=10 is given in Figure 2. It is easy to see that except for the main diagonal and half of the backward main diagonal, every cycle has exactly two of its elements in some superdiagonal positions. Within each cycle, the two elements in the superdiagonal positions are N/2 elements apart. Also, the cycle on the backward main diagonal has only N/2 elements and only one of them is in a superdiagonal position. Hence each permutation $\sigma A_r\sigma^t$ brings N-1 new elements to the superdiagonal for annihilation. The complexity of performing the permutation $\sigma A_r\sigma^t$ on a linearly connected network of N processing elements is also O(N), because there are N/2+1 cycles in the upper triangle of A and, as can be shown, six routing intructions are sufficient to rotate each cycle by one element.

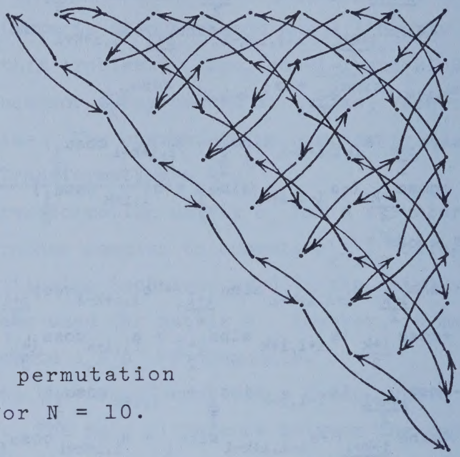


Figure 2. The permutation σ for N = 10.

5. THE COMPLEXITY OF A SWEEP

In the context of parallel processing, the definition of a sweep should be broadened to mean any finite sequence of orthogonal transformations that eliminates each of the N(N-1)/2 off-diagonal elements at least once. Based on this definition and on the fact that a transformation can eliminate at most N/2 off-diagonal elements, an algorithm would be optimal if it can complete a sweep in N-1 transformations.

In the present algorithm, every pair of consecutive $\phi A_r \phi^t$ and $\rho A_r \rho^t$ transformations eliminates N-1 superdiagonal elements and, using the permutation $\sigma A_r \sigma^t$, a sweep can be completed in N transformations and N/2 permutations. Hence the complexity of this algorithm is $O(N^2)$ per sweep. As a comparison, the K-S method requires $(N-1)^2$ transformations to complete a sweep, so its complexity is $O(N^3)$ per sweep.

6. SUMMARY

A practical and efficient Jacobi-like algorithm for computing the eigenvalues of real symmetric matrices has been presented. This algorithm uses the multiplication - by - diagonals technique for computing the transformations such that only a linearly connected network is needed for its parallel implementation. The storage scheme for the matrix on a parallel computer is simple and the interprocessor communication is regular. The complexity of the algorithm is $O(N^2)$ per sweep which is O(N) steps faster than the method by Kuch and Sameh.

REFERENCES

[1] K. Chen and K. Irani, "Mapping Problem and Graph Numbering," in Proc. of Workshop on Interconnection Network for parallel and distributed processing, Purdue University, April 21-22, 1980, pp. 41-46.

[2] D. Kuck and A. Sameh, "Parallel computation of eigenvalues of real matrices, "IFIP Congress 1971, North-Holland, Amsterdam, 1972 vol. 2, pp. 1266-1272.

[3] R. Gregory, "Computing eigenvalues and eigenvectors of a symmetric matrix on the Illiac", Mathematics of Computation 7 (1953), pp. 215-220.

[4] N. Madsen, G. Rodrigue and J. Karush, "Matrix multiplication by diagonals on a vector/parallel processor," Information Processing Letters, vol. 5, June 1976, pp. 41-45.

[5] H. Rutishauser, "The Jacobi method for real symmetric matrices", in Linear Algebra, (J. Wilkinson and C. Reinsch, editors), Spring-Verlag, N.Y. 1971, pp. 202-211.

[6] A. Sameh, "On Jacobi and Jacobi-like algorithms for a parallel computer", Mathematics of Computation, 25 (1971), pp. 579-596.

[7] J. H. Wilkinson, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.

APPENDIX

Given $B=\phi A\ \phi^t$, where $A=[a_{ij}]$ is real symmetric and ϕ is orthogonal and is defined by (1), (2), and (3), we express b_{ij} in terms of a_{ij} and t_{ij}.

Letting $T_k = \begin{pmatrix} t_{2k,2k} & t_{2k,2k+1} \\ t_{2k+1,2k} & t_{2k+1,2k+1} \end{pmatrix}$, we have the following:

$$\begin{aligned} b_{2i,2j} = {} & t_{2i,2i}(a_{2i,2j}t_{2j,2j}+a_{2i,2j+1}t_{2j,2j+1}) \\ & +t_{2i,2i+1}(a_{2i+1,2j}t_{2j,2j}+a_{2i+1,2j+1}t_{2j,2j+1}) \end{aligned}$$

$$\left\{\begin{aligned} b_{2i,2j+1} = {} & t_{2i,2i}(a_{2i,2j}t_{2j+1,2j}+a_{2i,2j+1}t_{2j+1,2j+1}) \\ & +t_{2i,2i+1}(a_{2i+1,2j}t_{2j+1,2j}+a_{2i+1,2j+1}t_{2j+1,2j+1}) \\ b_{2i+1,2j} = {} & t_{2i+1,2i}(a_{2i,2j}t_{2j,2j}+a_{2i,2j+1}t_{2j,2j+1}) \qquad (o \leq i,j \leq N/2-1) \end{aligned}\right.$$

$$\begin{aligned} & +t_{2i+1,2i+1}(a_{2i+1,2j}t_{2j,2j}+a_{2i+1,2j+1}t_{2j,2j+1}) \\ b_{2i+1,2j+1} = {} & t_{2i+1,2i}(a_{2i,2j}t_{2j+1,2j}+a_{2i,2j+1}t_{2j+12j+1}) \\ & +t_{2i+1,2i+1}(a_{2i+1,2j}t_{2j+1,2j}+a_{2i+1,2j+1}t_{2j+1,2j+1}) \end{aligned}$$

The algorithm in section 3.1 then follows since $t_{2k,2k} = t_{2k+1,2k+1} = \cos\theta_k$ and $t_{2k,2k+1} = -t_{2k+1,2k} = \sin\theta_k$.

EFFICIENT INVERSION FORMULAS FOR SUMS OF PRODUCTS OF TOEPLITZ AND HANKEL MATRICES *

B. FRIEDLANDER[1] and M. MORF[2]

[1] Systems Control, Inc.
1801 Page Mill Road
Palo Alto, CA 94304

[2] Information Systems Laboratory
Department of Electrical Engineering
Stanford University

ABSTRACT

A class of matrices that can be represented as sums of products of Toeplitz and Hankel matrices is considered. It is shown that the problem of inverting such matrices can be embedded in the problem of inverting a single block Toeplitz matrix, or a block α-Toeplitz matrix. Application of the multichannel Levinson algorithm, the generalized Levinson-Szego, or the recent doubling algorithms leads to the derivation of efficient inversion formulas for matrices in this class.

SUMMARY OF NOTATION

Define the i-th unit vector $e_i = [0, \dots, 0, 1, 0, \dots, 0]'$, the 1 being in the i-th place, the vector $x_{j:k} = \sum_{i=j}^{k} e_i x_i = [x_j', \dots, x_k']$, with the convention $x = x_{0:N}$, and the $N+1$ by $N+1$ lower shift matrix $Z = [e_2, \dots, e_{N+1}, 0]$, the generator for lower- and upper-triangular Toeplitz matrices

$$L(x) = [x, Zx, \dots, Z^N x] = \sum_{i=0}^{N} Z^i x_i = U'(x'), \; U(x) = [Z'^N x, \dots, Z'x, x] = L'(x').$$

The $N+1$ by $N+1$ anti-diagonal or "reversing order" matrix is defined as $\widetilde{I} = [e_{N+1}, \dots, e_1]$ and $\widetilde{M} \,\Delta\, M\widetilde{I}$ for any M. The lower and upper triangular Hankel matrices are now defined via column reversed Toeplitz equivalents

$$H^l(x) = L(x)\widetilde{I} = \widetilde{L}(x) \;, \quad H^u(x) = U(x)\widetilde{I} = \widetilde{U}(x) \quad \text{and}$$

$$\overline{x} = [0, x'_{0:N-1}]' \;, \quad \underline{x} = [x'_{1:N}, 0]' \;, \quad x = [\hat{x}', x'_N]' \;, \quad x = [x'_0, x^{v\prime}]' \;.$$

$$Q_{m,n} = [e_1, e_{1+n}, \dots, e_{1+(m-1)n} \mid e_2, e_{2+n}, \dots, e_{2+(m-1)n} \mid \dots, \mid e_n, e_{2n}, \dots, e_{mn}].$$

Note: $Q_{m,n}$ is a permutation matrix, hence orthogonal, i.e., $Q'_{m,n} Q_{m,n} = I$. It transforms a block matrix with mxm blocks of size nxn into a block matrix with nxn blocks of size mxm. Also note that $A \otimes B = Q_{m,n}(B \otimes A)Q'_{m,n}$, where A is m by m and B is n by n and $\otimes$ is the Kronecker product.

---

* Supported by the Defense Advanced Research Projects Agency Contract MDA903-80-C-0331.

1. INTRODUCTION

The problem of solving linear equations, or equivalently of inverting matrices, arises in many fields. Numerous techniques for doing this have been developed, but all of them have the characteristic that it takes in the order of N^3 operations* (multiplication and addition) to invert a general N x N matrix. A great deal of work has been devoted to finding more efficient ways of inverting matrices, by utilizing any special structure that they may posses. This special structure can take various forms, but there are many problems in mathematical physics and statistics in which the matrices can be shown to be related to Toeplitz or block Toeplitz, i.e.,

$$R = [r_{(i-j)}] \; , \quad 0 \leq i,j \leq N \; , \tag{1}$$

where r is a p x p block entry. Some examples for applications involving Toeplitz matrices are least squares estimation problems involving convolutions and integral equations with a difference kernel. Hankel matrices represent another type of special structure, characterized by

$$R = [r_{(i+j)}] \; , \quad 0 \leq i,j \leq N \; . \tag{2}$$

Several recursive algorithms have been developed for the inversion of Toeplitz and Hankel matrices, requiring only $O(N^2)$ arithmetic operations. See for example, [1] - [8]. The extension of this work to block-Toeplitz and block-Hankel matrices of size Np x Np (i.e., N x N blocks of size p x p), shows that the inverse can be obtained in $O(p^3N^2)$ operations [9] - [11]. See also [12] for generalization, and [13], [14] for a survey on inverses of Toeplitz operators. Recently some new algorithms have been developed that compute the inverse of a Toeplitz matrix even faster, in $O(N\log^2 N)$ operations, [15] - [17].

While Toeplitz and Hankel matrices appear in numerous applications, it often happens that practical problems involve matrices with a more complex structure. As an example, the inverse of a Toeplitz matrix is no longer Toeplitz but it retains a very simple structure. Other examples can be found in Section 2 and in [12], [18], [19]. In previous work [19], [20] we have shown that efficient inversion formulas can be obtained for matrices that are "close to Toeplitz" in some sense. More precisely, we considered the class of matrices represented by

$$T^{\alpha} = T + \sum_{i=1}^{\alpha} \sigma_i L_i U_i \; , \tag{3}$$

where T = full Toeplitz matrix, $L_i(U_i)$ = lower (upper) triangular Toeplitz matrix, $\sigma_i = \pm 1$ and α is the smallest integer for which this decomposition holds. We have shown that the number α associated with a given matrix is unique, and given by

---

* Strassen and others showed that fewer operations are required, i.e., $O(N^{2.78})$, see e.g., [17].

$$\alpha = rank\ \delta[T^{\alpha}] \ . \tag{4}$$

where $\delta[\cdot]$ is the shifted-difference operator given by

$$\delta[R] = [r_{i,j}]-[r_{i-1,j-1}] \ , \quad R = [r_{i,j}] \ , \quad 0 \leq i,j \leq N \ . \tag{5}$$

We will refer to a matrix of this type as having a shift-low rank [12] or a displacement rank of α [19], or more briefly, as an α-Toeplitz matrix. This class of matrices has a nice closure property in the sense that the inverse of α-Toeplitz matrix is also α-Toeplitz. For more details see [19], [21] and the discussion in Section 2.

The natural extension of these results to matrices that are "close to Hankel," involves the following type of representation:

$$H^{\alpha} = \overline{T}\tilde{I} + \sum_{i=1}^{\alpha} \overline{\sigma}_i \overline{L}_i \overline{U}_i \tilde{I} = T^{\alpha}\tilde{I} = \tilde{T}^{\alpha} \ , \tag{6}$$

where $\overline{T}, \overline{L}, \overline{U}, \overline{\sigma}$ are defined in the same way as T, L, U, σ. In other words, the α-Hankel matrix is simply a column-permuted version of an α-Toeplitz matrix. Thus, a trivial way of inverting a α-Hankel matrix is by transforming it into an α-Toeplitz matrix and then applying the algorithm given for instance in [19], i.e.,

$$[H^{\alpha}]^{-1} = \left[\overline{T} + \sum_{i=1}^{\alpha} \overline{\sigma}_i \overline{L}_i \overline{U}_i\right]^{-1} \tilde{I} \ . \tag{7}$$

However, it is also possible to invert the α-Hankel matrix directly, as described in [22]. The difference between these two types of inversion formulas lies in the sequence of submatrices that are recursively inverted. This subtle difference makes the trivial solution (7) inapplicable in many situations, e.g., if $H^{\alpha} = H^{u} \oplus H^{l}$.

In this paper we extend the class of matrices for which efficient inversion is possible, to include mixtures of Toeplitz and Hankel matrices. We consider matrices of the form

$$R = T + \sum_{i=1}^{\alpha} \sigma_i L_i U_i + \overline{T}\tilde{I} + \sum_{i=1}^{\beta} \overline{\sigma}_i \overline{L}_i \overline{U}_i \tilde{I} = T^{\alpha} + H^{\beta} \ , \tag{8}$$

i.e., sums of α-Toeplitz and β-Hankel matrices. This class of matrices has a closure property in the sense that R^{-1} has a representation of the type given in Equation (8). The possibility of this extension was already mentioned in [18], [19], [21], but the details were not presented. Applications of inversions of Toeplitz plus Hankel matrices are discussed in [23], [26].

The basic idea underlying our results, is to embed the α-Toeplitz/β-Hankel matrix in a larger matrix that has a structure for which efficient inversion formulas have already been developed. Two such embeddings are explored here. In

Section 3, we show that an α-Toeplitz/β-Hankel matrix R with p x p entries (i.e., $R = [r_{i,j}]$, $r_{i,j}$ is a p x p matrix), can be embedded in an $(\alpha+\beta)$-Toeplitz matrix with 2p x 2p blocks. Applying the inversion formulas derived in [19] to this matrix results in an algorithm which requires $O(4(\alpha + \beta + 2)p^2N^2)$ operations. In Section 4 we present a way of embedding the α-Toeplitz/β-Hankel matrix in a block Toeplitz matrix with $2(\alpha + \beta + 2)p \times 2(\alpha + \beta + 2)p$ entries. Applying the multichannel Levinson algorithm to this matrix requires $0(8(\alpha + \beta + 2)^3 p^3 N^2)$ operations. However, by using the special structure of the block entries the algorithm can be further simplified, resulting in an algorithm equivalent to the one presented in Section 3. Further reductions in computational requirements can be obtained by applying the class of doubling formulas (i.e., the $O(N \log^2 N)$ algorithms described in [15] - [17]) to the embedded block Toeplitz matrix.

2. AN ALGEBRA OF TOEPLITZ AND HANKEL MATRICES

Before presenting the main results of this paper, we will discuss in some detail the class of α-Toeplitz/β-Hankel matrices. This discussion is meant to provide some simple rules for manipulating products of Hankel and Toeplitz matrices. Such rules are very useful when one attempts to apply the results presented later to particular problems. First we will show that arbitrary sums of products of Toeplitz and Hankel matrices can be transformed to the form of Equation (8). To do this, note the following facts which follow directly from the definitions in the list of notation:

$$\begin{aligned} L(x)\tilde{I} &= H^{l}(x) = \tilde{L}(x)\,, \quad U(x)\tilde{I} = H^{u}(x) = \tilde{U}(x)\,, \\ \tilde{I}L(x) &= H^{u}(\tilde{I}x) = \tilde{U}(\tilde{x})\,, \quad \tilde{I}U(x) = H^{l}(\tilde{I}x) = \tilde{L}(\tilde{x})\,, \\ \tilde{I}L(x)\tilde{I} &= U(\tilde{I}x) = U(\tilde{x})\,, \quad \tilde{I}U(x)\tilde{I} = L(\tilde{I}x) = L(\tilde{x}). \end{aligned} \tag{9}$$

Another useful result is the fact that a product of upper- and lower-triangular Toeplitz matrices can be converted into lower-times upper-triangular type expressions via

$$U(x)\,L(y) = T - L(\bar{x})\,U(\underline{y}) \tag{10}$$

To see this, let $U^B = [U(x), L(\bar{x})]$, and $L^B = [L'(y), U'(\underline{y})]'$. It is easy to check that $U^B L^B$ is a Toeplitz matrix, i.e.,

$$T = U^B L^B = U(x)\,L(y) + L(\bar{x})\,U(\underline{y}) \tag{11}$$

Equation (10) follows from (11).

An arbitrary product of full Toeplitz or Hankel matrices can always be decomposed into a sum of products of triangular matrices. For example,

$$T \cdot H = (L + U)(H^{l} + H^{u}) = LH^{l} + UH^{l} + LH^{u} + UH^{u}\,,$$

where L, U are the lower- and upper-triangular parts of T (similarly for H^l, H^u). If it is desired to work with strictly lower (upper) triangular matrices, it is possible to write separately the diagonal part of T, i.e.,

$$T \cdot H = (t_0 I + L + U)(h_0 \tilde{I} + H^l + H^u) . \tag{13}$$

Table 1 summarizes all possible combinations of products of L, U, H^l and H^u. The main point illustrated by this table is the fact that any such product can be easily transformed into the form $T \pm LU$, $(\overline{T} \pm \overline{L}\,\overline{U})\tilde{I}$. Thus, an arbitrary sum of products of this type can always be transformed to the form of Equation (8).

TABLE 1

Type		Equivalent To
$L(x)L(y)$		$L(L(x)y)$
$L(x)U(y)$		$L(x)U(y)$
$L(x)H^u(y)$	$L(x)[H^u(y)\tilde{I}]\tilde{I}$	$L(x)U(y)\tilde{I}$
$L(x)H^\ell(y)$	$L(x)[H^\ell(y)\tilde{I}]\tilde{I} = L(x)L(y)\tilde{I}$	$L[L(x)y]\tilde{I}$
$U(x)L(x)$		$T - L(\bar{x})U(\underline{y})$
$U(x)U(y)$		$U[U(x)y]$
$U(x)H^u(y)$	$U(x)[H^u(y)\tilde{I}]\tilde{I} = U(x)U(y)\tilde{I}$	$U[U(x)y]\tilde{I}$
$U(x)H^\ell(y)$	$U(x)[H^\ell(y)\tilde{I}]\tilde{I} = U(x)L(y)\tilde{I}$	$[T - L(\bar{x})U(\underline{y})]\tilde{I}$
$H^u(x)L(y)$	$[H^u(x)\tilde{I}][\tilde{I}L(y)\tilde{I}]\tilde{I} = U(x)U(\tilde{I}y)\tilde{I}$	$U(U(x)\tilde{I}y)\tilde{I}$
$H^u(x)U(y)$	$[H^u(x)\tilde{I}][\tilde{I}U(y)\tilde{I}]\tilde{I} = U(x)L(\tilde{I}y)\tilde{I}$	$[\bar{T} - L(\bar{x})U(\underline{\tilde{I}y})]\tilde{I}$
$H^u(x)H^u(y)$	$[H^u(x)\tilde{I}][\tilde{I}H^u(y)] = U(x)L(\tilde{I}y)$	$[\bar{T} - L(\bar{x})U(\underline{\tilde{I}y})]$
$H^u(x)H^\ell(y)$	$[H^u(x)\tilde{I}][\tilde{I}H^\ell(y)] = U(x)U(\tilde{I}y)$	$U[U(x)\tilde{I}y]$
$H^\ell(x)L(y)$	$[H^\ell(x)\tilde{I}][\tilde{I}L(y)\tilde{I}]\tilde{I}$	$L(x)U(\tilde{I}y)\tilde{I}$
$H^\ell(x)U(y)$	$[H^\ell(x)\tilde{I}][\tilde{I}U(y)\tilde{I}]\tilde{I} = L(x)L(\tilde{I}y)\tilde{I}$	$L[L(x)\tilde{I}y]\tilde{I}$
$H^\ell(x)H^u(y)$	$[H^\ell(x)\tilde{I}][\tilde{I}H^u(y)] = L(x)L(\tilde{I}y)$	$L[L(x)\tilde{I}y]$
$H^\ell(x)H^\ell(y)$	$[H^\ell(x)\tilde{I}][\tilde{I}H^\ell(y)]$	$L(x)U(\tilde{I}y)$

Note: T is defined by Equation (11)

$\bar{T}$ is computed the same way as T, except that y is replaced by $\tilde{I}y = \tilde{y}$.

Products of more than two matrices of this type can also be reduced to the form of Equation (8). As an example consider

$$L(w)\,U(x)\,H^u(y)\,L(z) = L(w)\,U(x)(H^u(y)\,\tilde{I})(\tilde{I}\,L(z)\,\tilde{I})\tilde{I}$$

$$= L(w)\,U(x)\,U(y)\,U(\tilde{I}z)\tilde{I} = L(w)\,U(U(x)\,U(y)\,\tilde{I}z)\tilde{I}$$

$$= L(w)\,U(U(U(x)y)\,\tilde{I}z)\,\tilde{I}$$

Once a given mixture of Toeplitz and Hankel matrices is transformed into this standard form, there is still a question of the minimality of the representation. In other words, how to verify that the representation obtained as a result of these manipulations has the smallest number of terms. The answer lies, of course, in the notion of the displacement rank. It is easy to check (take a 3 x 3 case) that

$$\delta[L(x)\,U(y')] = x\,y' \;,\quad \delta[T] = 0 \;. \tag{14}$$

Thus, given a matrix $R = T_1 + \sum_{i=1}^{\alpha_1} L(x_i)\,U(y_i')$, we can check whether this is a minimal representation by looking at the displacement rank

$$\alpha \;\Delta\; rank\;\delta[R] = rank \sum_{i=1}^{\alpha_1} x_i\,y_i' \;. \tag{15}$$

If $\alpha = \alpha_1$, we know that this is a minimal representation. The proof is by contradiction: if there exists a representation of R with a smaller number of terms $\alpha_2 < \alpha$, say $R = T_2 + \sum_{i=1}^{\alpha_2} L(w_i)\,U(z_i')$, then $rank\;\delta[R] = rank \sum_{i=1}^{\alpha_2} w_i\,z_i'$ $\leqq \alpha_2 < \alpha$. If $\alpha < \alpha_1$, it is possible to find sets of vectors $\{w_i\}_{i=1}^{\alpha}$, $\{z_i\}_{i=1}^{\alpha}$ such that

$$\sum_{i=1}^{\alpha_1} x_i\,y_i' = \sum_{i=1}^{\alpha} w_i\,z_i' \;, \tag{16}$$

$$R = T + \sum_{i=1}^{\alpha} L(w_i)\,U(z_i') \;. \tag{17}$$

The sets $\{w_i\}$, $\{z_i\}$ can for instance be found using the Gram-Schmidt procedure on $\{x_i\}$, $\{y_i\}$ respectively.

3. THE α-TOEPLITZ EMBEDDING

Efficient inversion formulas for α-Toeplitz matrices were derived in [19] using the generalized Levinson-Szego recursions [20]. These formulas can be applied to the α-Toeplitz/β-Hankel case by using the following embedding. Let $R = T^{\alpha} + H^{\beta}$ as before. Note the following matrix identity,

$$\underset{\textstyle P}{\frac{1}{\sqrt{2}}\begin{bmatrix} I & \tilde{I} \\ -\tilde{I} & I \end{bmatrix}}\underset{\textstyle S}{\begin{bmatrix} T^{\alpha} & H^{\beta}\tilde{I} \\ \tilde{I}H^{\beta} & \tilde{I}T^{\alpha}\tilde{I} \end{bmatrix}}\underset{\textstyle P'}{\begin{bmatrix} I & -\tilde{I} \\ \tilde{I} & I \end{bmatrix}\frac{1}{\sqrt{2}}} = \begin{bmatrix} T^{\alpha} + H^{\beta} & 0 \\ 0 & \tilde{I}(T^{\alpha} - H^{\beta})\tilde{I} \end{bmatrix} . \tag{18}$$

The matrix P is orthogonal, i.e., $PP' = I$. The matrix S can be transformed into a block matrix (with 2p x 2p blocks) which has displacement $rank$ $(\alpha + \beta)$. In other words, there exist an interleaving permutation operator $Q_{2,N}$ such that

$$Q_{2,N}'\,S\,Q_{2,N} = \mathbf{T} + \sum_{i=1}^{\alpha} \mathbf{L}_i\,\mathbf{U}_i + \sum_{i=1}^{\beta} \overline{\mathbf{L}}_i\,\tilde{\mathbf{U}}_i \;, \tag{19}$$

where $\mathbf{T}$ is a block Toeplitz matrix with full blocks, $\mathbf{L}_i$, $\overline{\mathbf{L}}_i$, $\mathbf{U}_i'$ are block triangular Toeplitz matrices with diagonal blocks, and $\widetilde{U}_i$ is a block upper-triangular Toeplitz matrix with anti-diagonal blocks. The special structure of the blocks can be utilized to further reduce the number of computations involved in the inversion formula which is discussed next.

Using the results presented in [19] it can be shown, that the inverse of the permuted S matrix, an $(\alpha+\beta)$ - Toeplitz matrix has the following representation

$$[Q_{2,N}' \, S \, Q_{2,N}]^{-1} = T^* + \sum_{i=1}^{\alpha+\beta} L_i^* \, U_i^* \, . \tag{20}$$

The entries of T^*, L_i^*, U_i^* can be computed in $O(4(\alpha+\beta+2)p^2N^2)$ operations, using the generalized Levinson-Szego recursion. The full description of the algorithm is somewhat lengthy and is omitted here.

Once the inverse (20) is computed, we can recover the desired inverse $R^{-1} = (T^\alpha + H^\beta)^{-1}$ by

$$(P \, Q_{2,N})(Q_{2,N}' \, S \, Q_{2,N}')^{-1}(P \, Q_{2,N})' = \begin{bmatrix} (T^\alpha + H^\beta)^{-1} & 0 \\ 0 & \widetilde{I}(T^\alpha - H^\beta)^{-1}\widetilde{I} \end{bmatrix} \tag{21}$$

The effect of pre- and post-multiplying Equation (20) by $(P \, Q_{2,N})$ and $(P \, Q_{2,N})'$ respectively, causes $(T^\alpha + H^\beta)^{-1}$ to have the form (recall $\widetilde{T} = \overline{T\widetilde{I}}, \widetilde{U} = \overline{U\widetilde{I}}$)

$$(T^\alpha + H^\beta)^{-1} = T + \sum_{i=1}^{4(\alpha+\beta)} L_i \, U_i + T \, \widetilde{I} + \sum_{i=1}^{4(\alpha+\beta)} \overline{L}_i \, \overline{U}_i \, \widetilde{I} \, . \tag{22}$$

4. THE BLOCK TOEPLITZ EMBEDDING

The approach described in the previous section requires knowledge of the inversion formulas for block α-Toeplitz matrices. Here we describe a more direct approach which involves the inversion of block Toeplitz (rather than block α-Toeplitz) matrices. Efficient inversion of such matrices can be accomplished by the multichannel Levinson algorithm [9]-[12], rather than the generalized Levinson algorithm [19], [21], or by doubling algorithms [16], due to their relationship [24].

For the block Toeplitz embedding we need the well known matrix identity

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} [A - BD^{-1}C]^{-1} & -[A - BD^{-1}C]^{-1}BD^{-1} \\ -[D - CA^{-1}B]^{-1}CA^{-1} & [D - CA^{-1}B]^{-1} \end{bmatrix} . \tag{23}$$

REMARK: In the special case where $\alpha = \beta = 0$, $R = T + H$, i.e., the matrix R is a sum of a Toeplitz and a Hankel matrix. This case was discussed during conference talks, while presenting [16] and [18]. It is a little known fact that Levinson in [8] discusses the inversion of a Toeplitz matrix via the inversion of a Toeplitz plus a Hankel matrix of half size. The *inverse* idea, viz. embedding the Toeplitz plus Hankel matrix into a larger block Toeplitz matrix, stimulated the results on this paper. We note that [23] also discusses the special Toeplitz plus Hankel case, although unaware of [8].

where $A - BD^{-1}C$ is the Schur complement of D. Now consider the matrix

$$S = \begin{bmatrix} T+\widetilde{T} & L_{1:\alpha} & \overline{L}_{1:\beta} \\ U_{1:\alpha} & -\sigma_{1:\alpha} I & 0 \\ \overline{U}'_{1:\beta} & 0 & -\overline{\sigma}_{1:\beta}\widetilde{I} \end{bmatrix} \qquad (24)$$

Note that the Schur complement of of a submatrix of this matrix is given by

$$R = T + \sum_{i=1}^{\alpha} \sigma_i L_i U_i + \widetilde{T} + \sum_{i=1}^{\beta} \overline{\sigma}_i \overline{L}_i \widetilde{I} \overline{U}'_i ,$$

thus the (1,1) block of S^{-1} will be exactly the inverse matrix R^{-1}.

Next we decompose the matrix S into a Toeplitz part plus a Hankel part, a simple task, since all blocks are either Toeplitz or Hankel. Let $S = S_1 + S_2$, where S_1 equals the Toeplitz part of S, i.e., all the matrices with tildes are replaced by zeroes, and S_2 equals S with all entries zero, except for the matrices with tildes, i.e., Hankel matrices.

Note that the matrix S_1 contains $(\alpha + \beta + 1) \times (\alpha + \beta + 1)$ blocks with a Toeplitz structure. Thus, it is possible to transform the matrix into a block Toeplitz matrix. More precisely $Q'_{\alpha+\beta+1,N} S_1 Q_{\alpha+\beta+1,N} = [v_{i-j}] = V$, where all v_{i-j} have a very sparse structure, i.e., the diagonal blocks v_0 are diagonal themselves and the off-diagonal blocks $v_{i-j}, i \neq j$ have a nonzero first row or first column.

The matrix S_2 can be similarly transformed using the interleaving permutations on rows and columns $Q'_{(\alpha+\beta+1),N} S_2 Q_{(\alpha+\beta+1),N} = [\overline{v}_{i-j}] = \overline{V}$, where the anti-diagonal blocks are diagonal (with the leading $\alpha+1$ entries zero) and the rest of the blocks have only one non-zero entry in the $(\alpha+1, \alpha+1)$ position.

Thus, the matrix S has been transformed into a sum of a block Toeplitz and a block Hankel matrix. From the discussion in the previous section we already know that such a matrix can be embedded in a larger block Toeplitz matrix:

$$T = Q'P \begin{bmatrix} V & \overline{V} \\ \overline{V} & V' \end{bmatrix} P'Q = \text{block Toeplitz with } 2(\alpha+\beta+1) \times 2(\alpha+\beta+1) \text{ blocks} \qquad (25)$$

This block Toeplitz matrix can now be inverted[*] using the multichannel Levinson algorithm, see e.g., [9], [25], or via doubling type algorithms [16].

---

[*] We note that one has to make use of the fact that the inverse of a block Toeplitz matrix has the following representation, see e.g., [19], where $T = [t_{i-j}]$ is a block Toeplitz matrix with blocks t_{i-j} of size p x p. Then

$$T^{-1} = L(B')\, diag\{M_N^{-1}\}\, U(B) - L(A')\, diag\{N_N^{-1}\}\, U(A) ,$$

where A, B are defined by

$$\begin{bmatrix} I & B_{N,1}, & \cdots & B_{N,N} \\ A_{N,N} & \cdots & A_{N,1} & I \end{bmatrix} T = \begin{bmatrix} M_N & 0 & \cdots & 0 \\ 0 & \cdots & 0 & N_N \end{bmatrix}$$

Applying the Levinson type algorithm to T will require $O(8(\alpha + \beta + 1)^3 p^3 N^2)$. However, if the sparse structure of the blocks of T is fully utilized, the computational requirements can be reduced to at most $O(8(\alpha + \beta + 1)p^3 N^2)$. A similar reduction arises when doubling algorithms [16] are used.

The procedure desribed above can, of cource, be applied to α-Toeplitz matrices (i.e., the case where $H^{\beta} = 0$). In this case the second step of the embedding (Equation (29)) can be omittted, since $Q'^{S}_{(\alpha+2),N} Q_{(\alpha+2),N}$ is already a block Toeplitz matrix. This embedding combined with the multichannel Levinson algorithm provides a simple derivation of the generalized Levinson algorithm and the inversion formulas for α-Toeplitz matrices. The details of this approach can be found in [24]. In fact we can now interpret the two approaches presented in this paper as follows. To invert an α-Toeplitz/β-Hankel matrix, one can proceed in two ways. First embed the Toeplitz part and the Hankel part in an block $(\alpha + \beta)$-Toeplitz matrix. Then embed the block $\alpha + \beta)$-Toeplitz matrix in a block Toeplitz matrix. This is in effect what was done in Section 3. The second way first embeds the matrix R in a sum of block Toeplitz and block Hankel matrices, then the sum is embedded in a block Toeplitz matrix. This is the approach described in Section 4. These two approaches are equivalent provided that one takes into account the fine structure of the various block matrices involved.

5. CONCLUSIONS

The embedding approach presented in this paper makes it possible to use a class of efficient inversion algorithms, originally developed for Toeplitz matrices, for the much larger class of sums of products of Toeplitz and Hankel matrices. The emphasis of the previous sections was on the algebra of Toeplitz and Hankel matrices and the embedding technique, and not on explicit algorithmic results. Specific algorithms can be tailored for specific applications using these ideas. Several efficient computer routines are currentlly available for inverting block Toeplitz matrices. Combined with a program for performing the necessary data manipulation (e.g., column and row permutations) described in Sections 3 and 4, the same routines can be used for the α-Toeplitz/β-Hankel case. Alternatively, it is possible to write new inversion routines that operate directly on the given matrix (i.e., manipulate the equations of the Levinson algorithm rather than the input data). We also note that square-root implementations using orthogonal transformations (e.g., Householder type) or elementary rotations (e.g., Givens type) are numerically preferable, see e.g. [27].

The main power of the ideas presented here stems from embedding a given matrix with a seemingly complex structure into a larger (block) matrix, that has a simpler and recognizable structure. The embedding used here is generated by viewing the given matrix as the Schur complement of a submatrix of a larger block matrix with a simpler structure. The Schur complement is a rational combination of sufficient complexity to handle the matrices encountered in the case considered here. We may note that "higher-order" Schur complements can and often have to be used to accomplish this embedding, see e.g., [24]. Due to space limitation many details were omitted in this paper and will appear in a journal version.

REFERENCES

1. W.F. Trench, *J. Soc. Ind. & Appl. Math.*, Vol. 12, pp. 515-522, 1964.
2. W.F. Trench, *J. Soc. Ind. & Appl. Math.*, Vol. 13, pp. 1102-1107, 1965.
3. E.H. Bareiss, *Numerische Mathematik*, Vol. 13, pp. 404-424, 1969.
4. S. Zohar, *J. of the Association for Computing Machinery*, Vol. 16, pp. 592-601, 1969.
5. J.L. Phillips, *Mathematics of Computation*, Vol. 25, pp. 599-602, 1971.
6. J.H. Justice, *SIAM J. on Mathematical Analysis*, Vol. 5, pp. 503-508, 1974.
7. J. Rissanen, *Numerische Mathematik*, Vol 22, pp. 361-366, 1974.
8. N. Levinson, *J. Math. Phys.*, Vol. 25, pp. 261-278, January 1947.
9. R.A. Wiggins, E.A. Robinson, *J. Geoph.s.* Vol. 70, No. 8, pp. 1885-1891, 1965.
10. G.A. Watson, *J. Assoc. Comp. Mach.*, Vol. 20, pp. 409-415, 1973.
11. J. Rissanen, *Mathematics of Computation, Vol. 27, pp. 147- 154, 1973.*
12. M. Morf, *Fast Algorithms for Multivariable Systems*, Ph.D. Dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA,. 1974.
13. T. Kailath, A. Vieira and M. Morf, *SIAM Review*, Vol. 10, pp. 106-119, 1978.
14. U. Grenander and G. Szego, *Toeplitz Forms and Their Applications*, Univ. of California Press, Berkeley, 1958.
15. F.G. Gustavson and D.Y. Yun, "Fast Compuation of Pade Approximants and Toeplitz Systems of Equations Via the Extended Euclidean Algorithm," RC 7551, Mathematical Sciences Dept., IBM T.J. Watson Res. Ctr., Yorktown Heights, NY, 1979.
16. M. Morf, "Doubling Algorithms for Toeplitz and Related Equations," *Proc. Int. Conf. on ASSP*, pp. 954-959, April 1980. See also IEEE IT, to appear.
17. A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms* Addison-Wesley, Reading, MA, 1974.
18. M. Morf and T. Kailath, *Annals of Econ. & Soc. Meas.*, Vol. 6, No. 3, pp. 261-274, 1977.
19. B. Friedlander, M. Morf, T. Kailath, and L. Ljung, *Journal of Linear Algebra and Its Applications*, Elsevier North-Holland, Inc., vol. 27, pp. 31-60, October 1979
20. B. Friedlander, T. Kailath, M. Morf and L. Ljung, *IEEE Trans. Automatic Control*, AC-23, pp. 653-659, August 1978.
21. T. Kailath, S.-Y. Kung, M. Morf, *J. Math. Anal. Appl.*, Vol. 68, No. 2, April, 1979.
22. B. Friedlander and M. Morf, "Efficient Inversion of α-Hankel Matrices," submitted for pub.
23. G.A. Merchant and T.W. Parks, "Efficient Solution of A Toeplitz-Plus-Hankel Coefficient Matrix System of Equations," submitted for publication.
24. J.M. Delosme and M. Morf, "A Tree Classification of Algorithms for Toeplitz and Related Equations Including Generalized Levinson and Doubling Type Algorithms," *Proc. 19th IEEE CDC*, December 10-12, 1980. (to appear)
25. T. Kailath, *IEEE Trans. Inform. Theory*, Vol. IT-20, pp. 146-181, 1974.
26. B. Friedlander and M. Morf, "Efficient Algorithms for Adaptive Linear-Phase Filter," submitted to ICASSP, 1981.
27. M. Morf, C. Muravchik and D.T.L. Lee, "Hilbert Space Array Methods for Alpha Stationary Processes," submitted to ICASSP, 1981.

VARIABLE METRIC METHODS FOR CONSTRAINED OPTIMIZATION USING DIFFERENTIABLE EXACT PENALTY FUNCTIONS*

DIMITRI P. BERTSEKAS
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

This paper relates to the application of variable metric methods for solving twice continuously differentiable equality constrained optimization problems. We show that the region of convergence of such methods can be enlarged by making use of differentiable exact penalty functions due to DiPillo and Grippo [1], and Fletcher [2]. The methods are well suited for use in conjunction with Powell's variable metric formula [3] and bypass some of the inherent disadvantages of nondifferentiable exact penalty functions.

1. INTRODUCTION

We consider the constrained optimization problem

(ECP) minimize $f(x)$
subject to $h(x) = 0$

where $f : R^n \to R$ and $h : R^n \to R^m$ are three times continuously differentiable functions. We focus primary attention at local minima-Lagrange multiplier pairs (x^*,λ^*) satisfying the following second order sufficiency conditions for optimality

$$\nabla_x L(x^*,\lambda^*) = 0, \; h(x^*) = 0 \tag{1}$$

$$x^* \varepsilon X^* \tag{2}$$

$$z'\nabla^2_{xx} L(x^*,\lambda^*) z > 0, \; \forall z \neq 0, \; \nabla h(x^*)'z = 0, \tag{3}$$

where $L : R^{n+m} \to R$ is the Lagrangian function

$$L(x,\lambda) = f(x) + \lambda' h(x) \tag{4}$$

and X^* is the set given by

$$X^* = \{x | \nabla h(x) \text{ has rank } m\}. \tag{5}$$

The methods to be considered are of the form

$$x_{k+1} = x_k + \alpha_k \Delta x_k \tag{6}$$

---

*Work supported by NSF Grant ENG-79-06332

where α_k is a positive stepsize parameter, and Δx_k together with some vector λ_{k+1} solves a system of the form

$$\begin{bmatrix} H_k & \nabla h(x_k) \\ \nabla h(x_k)' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) \\ h(x_k) \end{bmatrix}. \tag{7}$$

The only requirements are that $x_k \varepsilon X^*$ and that H_k is a symmetric n x n matrix which is positive definite on the tangent plane

$$T_k = \{z | \nabla h(x_k)'z = 0\}, \tag{8}$$

i.e.,

$$z'H_k z > 0, \quad \forall z \neq 0, \ z \varepsilon T_k. \tag{9}$$

When a unity stepsize is used in (6), and $H_k = \nabla^2_{xx} L(x_k,\lambda_k)$ the method reduces to Newton's method for solving the system of necessary conditions

$$\nabla_x L(x,\lambda) = 0, \quad h(x) = 0. \tag{10}$$

It is well known that, when the starting point (x_o,λ_o) is sufficiently close to a pair (x^*,λ^*) satisfying (1)-(3), this method converges to (x^*,λ^*) superlinearly. Recent work has been directed towards bypassing the need for computing the Hessian $\nabla^2_{xx} L$ by employing Quasi-Newton approximations, and the need for a good starting point by means of a stepsize procedure involving descent of some exact penalty function. Much of the research that has been done is based on the use of the exact penalty function

$$f(x) + c \max\{|h_1(x)|,\ldots,|h_m(x)|\} \tag{11}$$

where c is a positive scalar parameter and the works of Powell [3],[4], and Mayne and Polak [5] are characteristic in this respect. This paper utilizes instead the exact penalty function

$$\hat{P}_\tau(x;c) = \min_\lambda P_\tau(x,\lambda;c) \tag{12}$$

where

$$P_\tau(x,\lambda;c) = L(x,\lambda) + \frac{c+\tau|\lambda|^2}{2} |h(x)|^2 + \frac{1}{2} |M(x)\nabla_x L(x,\lambda)|^2. \tag{13}$$

Here $\tau \geq 0$ is a fixed scalar and M(x) is an m x n twice continuously differentiable matrix function on X* such that $M(x)\nabla h(x)$ is nonsingular for all $x \varepsilon X^*$. The penalty function (13) is due to DiPillo and Grippo [1] who considered the case $\tau = 0$. The penalty function (12) is strongly related to the class of functions introduced by Fletcher [2] as explained in Bertsekas [6]. For example if

$$M(x) = [\nabla h(x)'\nabla h(x)]^{-1}\ \nabla h(x)'$$

then, using the fact that P_τ is positive definite quadratic in λ, the minimization in (12) can be carried out explicitly to yield

$$\hat{\lambda}(x) = \frac{1}{1+\tau|h(x)|^2}\ [\lambda(x)-h(x)]$$

where

$$\lambda(x) = -\ [\nabla h(x)'\nabla h(x)]^{-1}\ \nabla h(x)'\nabla f(x).$$

If $\tau = 0$, by substitution in (13) we obtain

$$\hat{P}_o(x;c) = L[x,\lambda(x)] + \frac{c-1}{2}\ |h(x)|^2$$

which is a primary example of Fletcher's class [2]. A positive value of τ is beneficial for computational purposes and this is the main reason for introducing it (see DiPillo, Grippo and Lampariello [7] and Bertsekas [6]).

The main property of the penalty function $\hat{P}_\tau$ is that, for c sufficiently large, a vector x* satisfying (1)-(3) is a strong local minimum of $\hat{P}_\tau$ as shown by Fletcher [2] for special cases, and more generally by Bertsekas [14].

In this paper we establish that the direction Δx_k generated by solving the system (7) is a descent direction of the penalty function $\hat{P}_\tau(\cdot;c)$ provided c is sufficiently large. We then consider the case where the condition

$$[H_k - \nabla^2_{xx}L(x^*,\lambda^*)]Z^* \to 0$$

is satisfied. Here Z* is an n x(n-m) matrix the columns of which span the tangent plane

$$T^* = \{z|\nabla h(x^*)'z = 0\}.$$

We show under these circumstances that, if $\alpha_k = 1$ for all k sufficiently large, the convergence to x* is superlinear. We also show that sufficiently close to x* the stepsize $\alpha_k = 1$ is acceptable in the sense that it reduces the value of $\hat{P}_\tau$. These results lead to an algorithm employing a Quasi-Newton updating formula for H_k satisfying $[H_k - \nabla^2_{xx}L(x^*,\lambda^*)]Z^* \to 0$, such as the one of Powell [3], and line search by descent on $\hat{P}_\tau$, with typically superlinear convergence rate. By contrast [8] a result of this type is impossible to demonstrate for the same algorithm when used in conjunction with the nondifferentiable penalty function (11). While there are methods for bypassing this difficulty ([5],[9]), at least at the theoretical level, one must view this feature of the nondifferentiable penalty function (11) as rather unfortunate.

The main results of the paper are presented in Section 2. These results are new but descend naturally from the author's earlier work [6] on a

Newton-like version of the algorithm considered here which in turn is related to locally convergent algorithms of Pschenichny and Danilin [10], and Tapia [11]. We mention also related work of Dixon [12] who proposed a variable metric algorithm based on descent of the function $P_o(\cdot,\cdot;c)$ with $M(x) = \nabla h(x)'$.

The notation employed in the paper is as follows. All vectors are finite dimensional and will be considered to be column vectors. A prime denotes transposition. The usual norm on the Euclidean space R^n is denoted by $|\cdot|$ [i.e., $|x| = (x'x)^{1/2}$ for all $x \varepsilon R^n$]. For a mapping $h\colon R^n \to R^m$, $h = (h_1,\ldots,h_m)'$ we denote by $\nabla h(x)$ the $n \times m$ matrix with columns the gradients $\nabla h_1(x),\ldots,\nabla h_m(x)$. Whenever there is danger of confusion we explicitly indicate the argument of differentiation--for example $\nabla_x L(x^*,\lambda^*)$ denotes gradient of L with respect to x evaluated at the pair (x^*,λ^*).

2. CONVERGENCE AND SUPERLINEAR RATE OF CONVERGENCE ANALYSIS

Consider the penalty function

$$\hat{P}_\tau(x;c) = \min_\lambda P_\tau(x,\lambda;c)$$

where

$$P_\tau(x,\lambda;c) = L(x,\lambda) + \frac{c+\tau|\lambda|^2}{2}\,|h(x)|^2 + \frac{1}{2}\,|M(x)\ \nabla_x L(x,\lambda)|^2$$

and $M(\cdot)$ is an $m \times n$ twice continuously differentiable matrix function such that $M(x)\nabla h(x)$ is nonsingular for all x in the set X^* of (5). Since P_τ is positive definite quadratic in λ, the minimizing vector $\hat{\lambda}(x)$ can be found analytically by solving the equation

$$0 = \nabla_\lambda P_\tau(x,\lambda;c) = h(x) + \tau|h(x)|^2\lambda + \nabla h(x)'M(x)'M(x)\nabla_x L(x,\lambda). \tag{14}$$

We thus obtain the minimizing vector

$$\hat{\lambda}(x) = -[\nabla h(x)'M(x)'M(x)\nabla h(x) + \tau|h(x)|^2 I]^{-1}[h(x) + \nabla h(x)'M(x)'M(x)\nabla f(x)] \tag{15}$$

and the equation

$$\hat{P}_\tau(x;c) = P_\tau[x,\hat{\lambda}(x);c], \qquad \forall\, x\varepsilon X^*. \tag{16}$$

Differentiating this equation and taking into account (14) we obtain

$$\begin{aligned}\nabla\hat{P}_\tau(x;c) &= \nabla_x P_\tau[x,\hat{\lambda}(x);c]\\ &= \nabla_x L[x,\hat{\lambda}(x)] + (c + \tau|\hat{\lambda}(x)|^2)\nabla h(x)h(x) + Q(x)M(x)\nabla_x L[x,\hat{\lambda}(x)]\end{aligned} \tag{17}$$

where

$$Q(x) = \nabla_x[M(x)\nabla_x L[x,\hat{\lambda}(x)]].$$

The following proposition shows that the vector Δx_k obtained by solving system (7) is a descent direction of $\hat{P}_\tau$ over an arbitrary compact subset of X^* provided c is sufficiently large.

Proposition 1: Let H be a compact set of symmetric n x n matrices, let b,B be two positive scalars with $b < B$, and let X be a compact subset of X^*. There exist scalars $\bar{c} > 0$ and $w > 0$ (depending on H, b, B, and X) such that for every $x \varepsilon X$ and matrix $H \varepsilon H$ satisfying

$$b|z|^2 \leq z'Hz \leq B|z|^2, \ \forall z \varepsilon R^n \text{ with } \nabla h(x)'z = 0, \tag{18}$$

the solution $(\Delta x,\lambda)$ of the system

$$\begin{bmatrix} H & \nabla h(x) \\ \nabla h(x)' & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \lambda \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ h(x) \end{bmatrix} \tag{19}$$

exists, is unique and satisfies for all $c \geq \bar{c}$

$$\nabla\hat{P}_\tau(x;c)'\,\Delta x \leq - w|\nabla\hat{P}_\tau(x;c)|^2. \tag{20}$$

Proof: For $x\varepsilon X$ let us use the abbreviated notation

$$\nabla_x L[x,\hat{\lambda}(x)] = \alpha, \quad h(x) = \beta, \quad \nabla h(x) = N \tag{21}$$

$$M(x) = M, \quad Q(x) = Q. \tag{22}$$

We will assume without loss of generality that for all $x\varepsilon X$ and $H\varepsilon H$ satisfying (18), the matrix H is positive definite with eigenvalues uniformly bounded above and bounded away from zero, since if we replace H by $H + pNN'$ where p is a sufficiently large scalar, then we will obtain a positive definite matrix without affecting either (18) or the solution Δx of system (19). (See also [14] for a more detailed explanation of this point.)

We will now evaluate the inner product

$$J = \nabla\hat{P}_\tau(x;c)'\Delta x. \tag{23}$$

Using (17) we have

$$J = \alpha'\Delta x + \alpha'M'Q'\Delta x + (c + \tau|\hat{\lambda}(x)|^2)\beta'N'\Delta x. \tag{24}$$

From (19) we have

$$N'\Delta x = -\beta \tag{25}$$

and a straightforward calculation shows that

$$\Delta x = = [H^{-1} - H^{-1}N(N'H^{-1}N)^{-1}N'H^{-1}]\alpha - H^{-1}N(N'H^{-1}N)^{-1}\beta. \qquad (26)$$

From (14) we also obtain

$$M\alpha = -(N'M')^{-1}[I + \tau\hat{\lambda}(x)\beta']\beta. \qquad (27)$$

from which

$$\alpha'M'M\alpha = \beta'[I + \tau\beta\hat{\lambda}(x)'](MN)^{-1}(N'M')^{-1}[I + \tau\hat{\lambda}(x)\beta']\beta. \qquad (28)$$

Substituting (25), (26), (27) and using (28) in (24) we find that J can be expressed as the sum of two quadratic forms

$$J = R(\alpha,\beta) - c|\beta|^2 \qquad (29)$$

where

$$\begin{aligned} R(\alpha,\beta) = &- \alpha'[H^{-1} - H^{-1}N(N'H^{-1}N)^{-1}N'H^{-1} + M'M]\alpha \\ &- \alpha'H^{-1}N(N'H^{-1}N)^{-1}\beta \\ &+ \beta'[I + \tau\beta\hat{\lambda}(x)'](MN)^{-1}Q[H^{-1} - H^{-1}N(N'H^{-1}N)^{-1}N'H^{-1}]\alpha \\ &- \beta'\{\tau|\hat{\lambda}(x)|^2 I - [I + \tau\beta\hat{\lambda}(x)'](MN)^{-1} \\ &\quad \{Q'H^{-1}N(N'H^{-1}N)^{-1} + (N'M')^{-1}[I + \tau\hat{\lambda}(x)\beta']\}\}\beta. \end{aligned}$$

The quadratic form $R(\alpha,\beta)$ when restricted on the subspace $\{(\alpha,\beta)|\beta = 0\}$ can be written

$$\begin{aligned} R(\alpha,0) &= - \alpha'[H^{-1} - H^{-1}(N(N'H^{-1}N)^{-1}N'H^{-1} + M'M]\alpha \\ &= - \bar{\alpha}'[I - \bar{N}(\bar{N}'\bar{N})^{-1}\bar{N}' + \bar{M}'\bar{M}]\bar{\alpha} \end{aligned}$$

where

$$\bar{\alpha} = H^{-\frac{1}{2}}\alpha, \; N = H^{-\frac{1}{2}}N, \; M = MH^{\frac{1}{2}}$$

We claim that $R(\alpha,0)$ is negative definite. Indeed since both matrices $[I - \bar{N}(\bar{N}'\bar{N})^{-1}\bar{N}']$ and $\bar{M}'\bar{M}$ are positive semidefinite it follows that $R(\alpha,0)$ is negative semidefinite. If $R(\bar{\alpha},0) = 0$ for some $\bar{\alpha} \neq 0$ then we must have $\bar{\alpha}'[I - \bar{N}(\bar{N}'\bar{N})^{-1}\bar{N}']\bar{\alpha} = 0$ and $\overline{M\alpha} = 0$. The first equation implies that $\bar{\alpha} = \bar{N}\xi$ for some $\xi\varepsilon R^m$ and the second equation implies $0 = \overline{MN}\xi = MN\xi$. Hence $\xi = 0$ and it follows that $\bar{\alpha} = 0$ which is a contradiction. Having established negative definiteness of $R(\alpha,0)$ it follows using (29) and a standard result that, for sufficiently large c, J can be represented as a negative definite quadratic form, i.e.

$$J = [\alpha' \ \beta']D(x,H,c)\begin{bmatrix}\alpha \\ \beta\end{bmatrix}$$

where $D(x,H,c)$ is a negative definite matrix. This argument can be easily extended to show that there exists $\bar{c} > 0$ such that for all $c \geq \bar{c}$, $x \varepsilon X$ and $H \varepsilon \mathcal{H}$ satisfying (18) the eigenvalues of $D(x,H,c)$ are negative and uniformly bounded below and away from zero. Since, in view of (17), the square norm of the gradient $\nabla \hat{P}_\tau(x;c)$ can also be expressed as a quadratic form in (α,β), it follows that there exists $w > 0$ such that (20) holds. QED

Proposition 1 shows that the algorithm

$$x_{k+1} = x_k + \alpha_k \Delta x_k,$$

where Δx_k is obtained by solution of system (7) and α_k is obtained by line search on the function $\hat{P}_\tau$, has global convergence properties provided c is chosen sufficiently large. We will now make the assumption that $\{H_k\}$ is bounded, the algorithm converges to a local minimum x^* satisfying together with a Lagrange multiplier λ^* the sufficiency conditions (1)-(3), and the condition

$$[H_k - \nabla^2_{xx} L(x^*,\lambda^*)]Z^* \to 0 \tag{30}$$

holds where Z^* is an $n \times (n-m)$ matrix the columns of which form a basis for the tangent plane

$$T^* = \{z \mid h(x^*)'z = 0\}.$$

We will show that for sufficiently large k, the stepsize $\alpha_k = 1$ is acceptable (in the sense that it leads to a reduction of $\hat{P}_\tau$ which is sufficient to enforce convergence) and the rate of convergence is superlinear. To this end it will be sufficient to show (compare with standard proofs of rate of convergence of Newton's method [13]) that (30) implies

$$\nabla^2 \hat{P}_\tau(x^*;c)\Delta x_k = -\nabla \hat{P}_\tau(x_k;c) + o(|x_k - x^*|). \tag{31}$$

We will show (31) for the special case

$$M(x) = [\nabla h(x)'\nabla h(x)]^{-1}\nabla h(x)', \qquad \tau = 0 \tag{32}$$

The details for the general case are similar but quite messy. From (15) we have

$$\hat{\lambda}(x) = -h(x) - [\nabla h(x)'\nabla h(x)]^{-1}\nabla h(x)'\nabla f(x) \tag{33}$$

and substitution in (16) yields

$$\hat{P}_0(x;c) = L[x,\hat{\lambda}(x)] + \frac{c+1}{2}|h(x)|^2.$$

We have by differentiation

$$\nabla\hat{P}_0(x;c) = \nabla_x L[x,\hat{\lambda}(x)] + \nabla\hat{\lambda}(x)h(x) + (c+1)\nabla h(x)h(x)$$

$$\nabla^2\hat{P}_0(x^*;c) = \nabla^2_{xx}L(x^*,\lambda^*) + \nabla\hat{\lambda}(x^*)\nabla h(x^*)' + \nabla h(x^*)\nabla\hat{\lambda}(x^*)'$$
$$+ (c+1)\nabla h(x^*)\nabla h(x^*)'.$$

Equation (31) can thus be written as

$$[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla\hat{\lambda}(x^*)\nabla h(x^*)' + \nabla h(x^*)\nabla\hat{\lambda}(x^*)' + (c+1)\nabla h(x^*)\nabla h(x^*)']\Delta x_k$$
$$= -[\nabla_x L[x_k,\hat{\lambda}(x_k)] + \nabla\hat{\lambda}(x_k)h(x_k) + (c+1)\nabla h(x_k)h(x_k)]$$
$$+ o(|x_k - x^*|). \tag{34}$$

We have (using the boundedness of $\{H_k\}$)

$$\Delta x_k = O(|\nabla_x L[x_k,\hat{\lambda}(x_k)]|) + O(|h(x_k)|) = O(|x_k - x^*|) \tag{35}$$

$$\nabla h(x_k)'\Delta x_k = -h(x_k). \tag{36}$$

Using these equations we obtain

$$\nabla\hat{\lambda}(x_k)h(x_k) = -\nabla\hat{\lambda}(x_k)\nabla h(x_k)'\Delta x_k = -\nabla\hat{\lambda}(x^*)\nabla h(x^*)'\Delta x_k + o(|x_k - x^*|) \tag{37}$$

$$\nabla h(x_k)h(x_k) = -\nabla h(x_k)\nabla h(x_k)'\Delta x_k = -\nabla h(x^*)\nabla h(x^*)'\Delta x_k + o(|x_k - x^*|). \tag{38}$$

Combining (34), (37) and (38) we see that (31) is equivalent to

$$[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla h(x^*)\nabla\hat{\lambda}(x^*)']\Delta x_k + \nabla_x L[x_k,\hat{\lambda}(x_k)] = o(|x_k - x^*|). \tag{39}$$

From the definition of $\hat{\lambda}(\cdot)$ we have for all $x \varepsilon X^*$

$$\nabla h(x)'\nabla_x L[x,\hat{\lambda}(x)] = -[\nabla h(x)'\nabla h(x)]h(x). \tag{40}$$

Since $\nabla_x L[x_k,\hat{\lambda}(x_k)] = O(|x_k - x^*|)$ we obtain from (40)

$$\nabla h(x^*)'\nabla_x L[x_k,\hat{\lambda}(x_k)] = -[\nabla h(x_k)'\nabla h(x_k)]h(x_k) + o(|x_k - x^*|). \tag{41}$$

Also by differentiating (40) at x* we obtain

$$\nabla h(x^*)'[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla h(x^*)\nabla\hat{\lambda}(x^*)'] = -[\nabla h(x^*)'\nabla h(x^*)]\nabla h(x^*)'$$
$$= -[\nabla h(x_k)'\nabla h(x_k)]\nabla h(x_k)' + O(|x_k - x^*|). \tag{42}$$

Combining (41), (42) and using (35), (36) we obtain

$$\nabla h(x^*)'\{[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla h(x^*)\nabla\hat{\lambda}(x^*)']\Delta x_k + \nabla_x L[x_k,\hat{\lambda}(x_k)]\} = o(|x_k - x^*|). \tag{43}$$

We will now show that

$$Z^{*\prime}\{[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla h(x^*)\nabla\hat{\lambda}(x^*)']\Delta x_k + \nabla_x L[x_k,\hat{\lambda}(x_k)]\} = o(|x_k - x^*|). \tag{44}$$

Since the n x n matrix

$$[\nabla h(x^*)\ Z^*]$$

is invertible, equations (43), (44) imply the desired relation (39) and hence also (31). In order to show (44) we note that the condition (30), equ. (35), and the fact $Z^{*\prime}\nabla h(x^*) = 0$ imply

$$Z^{*\prime}[\nabla^2_{xx}L(x^*,\lambda^*) + \nabla h(x^*)\nabla\hat{\lambda}(x^*)']\Delta x_k = Z^{*\prime}H_k\Delta x_k + o(|x_k - x^*|). \tag{45}$$

We also have from the definition of Δx_k

$$H_k\Delta x_k + \nabla h(x_k)[\lambda_{k+1} - \hat{\lambda}(x_k)] = -\nabla_x L[x_k,\hat{\lambda}(x_k)], \tag{46}$$

where λ_{k+1} is obtained from the solution of system (7). It is easily seen that $\lambda_{k+1} - \hat{\lambda}(x_k) = 0(|x_k - x^*|)$ so by using also the fact $Z^{*\prime}\nabla h(x_k) = 0(|x_k - x^*|)$, we obtain from (46)

$$Z^{*\prime}H_k\Delta x_k + Z^{*\prime}\nabla_x L[x_k,\hat{\lambda}(x_k)] = o(|x_k - x^*|). \tag{47}$$

By combining (45) and (47) we see that (44) holds and therefore our proof of validity of (31) is complete.

To summarize, we have shown that when the algorithm is coupled with a stepsize rule (such as the Armijo rule) that uses a unity initial stepsize and converges to a local minimum x* satisfying the sufficiency conditions (1)-(3) a superlinear rate of convergence is obtained provided the following two conditions hold:

a) The penalty parameter c is sufficiently large to ensure that x* is a strong local minimum of $\hat{P}_\tau$.

b) The sequence $\{H_k\}$ is bounded and

$$[H_k - \nabla^2_{xx}L(x^*,\lambda^*)]Z^* \to 0.$$

REFERENCES

[1] DiPillo G., and Grippo, L., "A New Class of Augmented Lagrangians in Nonlinear Programming", SIAM J. Control and Optimization, Vol. 17, 1979, pp. 618-628.

[2] Fletcher, R., "A Class of Methods for Nonlinear Programming with Termination and Convergence Properties", in Integer and Nonlinear Programming, J. Abadie (ed.), North-Holland, Amsterdam, 1970.

[3] Powell, M.J.D., "Algorithms for Nonlinear Constraints that use Lagrangian Functions", Math. Programming, Vol. 14, 1978, pp. 224-248.

[4] Powell, M.J.D., "The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations", in Nonlinear Programming 3, O.L. Mangasarian, R. Meyer, and S. Robinson (eds.), Academic Press, 1978, pp. 144-157.

[5] Mayne, D.Q., and Polak, E., "A Superlinearly Convergent Algorithm for Constrained Optimization Problems", Research Report 78/52, Department of Computing and Control, Imperial College, 1978.

[6] Bertsekas, D.P., "Enlarging the Region of Convergence of Newton's Method for Constrained Optimization", LIDS-R-985, Mass. Institute of Technology, March 1980, to appear in J.O.T.A.

[7] DiPillo, G., Grippo, L., and Lampariello, F., "A Method for Solving Equality Constrained Optimization Problems by Unconstrained Minimization", Proc. 9th IFIP Conference on Optimization Techniques, Warsaw, 1979.

[8] Maratos, N., "Exact Penalty Function Algorithms for Finite Dimensional and Control Optimization Problems", Ph.D. Thesis, Imperial College of Science and Technology, University of London, 1978.

[9] Chamberlain, R.M., Lemarechal, C., Pedersen, H.C., and Powell, M.J.D., "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization", Presented at the Tenth International Symposium on Math. Programming, Montreal, 1979.

[10] Pschenichny, B.N., and Danilin, Y.M., Numerical Methods in Extremal Problems, M.I.R. Publishers, Moscow, 1975 (English Translation 1978).

[11] Tapia, R.A., "Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization", J.O.T.A., Vol. 22, 1977, pp. 135-194.

[12] Dixon, L.C.W., "On the Convergence Properties of Variable Metric Recursive Quadratic Programming Methods", Numerical Optimization Centre, The Hatfield Polytechnic, March 1980.

[13] Ortega, J.M., and Reinboldt, W.C., Iterative Solution on Nonlinear Equations in Several Variables, Academic Press, N.Y., 1970.

[14] Bertsekas, D.P., Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 1981 (to appear).

CONTROL AND COMPUTING FOR AUTOMATED INTENSIVE CARE OF THE CARDIOVASCULAR SYSTEM

N. DeCLARIS AND K.M. KEMPNER
Maryland Institute for Emergency Medical Services Systems and the
College of Engineering
University of Maryland

ABSTRACT

A state-space approach to the automated management of the cardiovascular system in critically ill patients can be made compatible with critical care medical practice. The formulation leads to a configuration that can accommodate several drugs, fluids and therapeutic interventions.

SUMMARY

Patient management following severe trauma and/or shock involves the use of fluids, drugs and therapeutic interventions which rapidly affect the cardiovascular system. The use of these agents require continuous monitoring their effect and frequently adjusting the administration rate.(1) Recently computers have been used in closed-loop configurations to initiate and automatically adjust dose rates. The computer (mini or micro) receives on-line inputs from patient monitoring instrumentation and provides on-line outputs to adjust the rate of the therapeutic agents. From the system theoretic viewpoint such a configuration is a computer-based implementation of an interactive algorithm with important computational (complexity, speed) and interactive (with physicians, nurses) aspects. A deep analysis of the engineering and medical considerations of the problem in fact identifies that automated management of the critically ill patient involves

I. An algorithm to control the rate of the therapeutic agents
II. An algorithm to assess the current state of the patient
III. An algorithm to decide the medical protocol (management)

The complexity of each algorithm, if it is to be computer-implemented, depends on the extent of physician-interaction it requires. Viewed in this light, the computer based closed-loop configurations that have appeared in published literature todate utilize the greatest part of the computer resources for bookkeeping purposes and for controlling the rate of drug administration. Very little, if any, machine automation is carried out in patient assessment and in medical protocol decision functions - these are drawn upon from clinical experience and applied to the particular patient. However we have found that a suitable state-space formulation of the cardiovascular system,(2) which includes pharmaco-kinetics and transducer dynamics, reveals a broad class of parameter estimation algorithms on the basis of which patient-status may be automatically assessed by the computer on-line. This is possible only when the dimensionality of the state-space is increased manyfold; in addition the system dynamics become highly nonlinear. However these computational difficulties may be compensated by the recent advances in the throughput of the new generation of microprocessors (like the Z-8000). Another feature of this enhanced state-space approach is optimization of the medical protocol decision pathways based on data obtained from similarly automated past patients. Implementation requires a suitable computer-network structure or a large size multiple access computer. From these two alternatives the first one is much more attractive.

REFERENCES

1. N. DeClaris, R. A. Cowley and B. J. Trump "A Systems Approach to Intensive Care Medicine" The American Surgeon, Feb. 1979, p. 86-92.
2. N. DeClaris and K. M. Kempner to be published.

GENERAL COMPOSITE CONTROL LAWS FOR NONLINEAR SINGULARLY PERTURBED SYSTEMS

M. SUZUKI
Department of Aeronautical Engineering
Nagoya University
Chikusa-ku, Nagoya 464, JAPAN

ABSTRACT

The controllability and the stabilizability invariance of the reduced system (slow subsystem) of the nonlinear singularly perturbed system due to a class of fast feedback controls is shown. The general type composite control for the nonlinear singularly perturbed system is proposed and its application for the synthesis problem is considered.

1. INTRODUCTION

The singularly perturbed system is defined mathematically as the system including the singular parameters whose small perturbation can change the order of the system. The presence of such parameters often makes the analysis of the original system difficult due to the facts that the system has a singularity and that the presence of singular parameters increase the dimensionality of the system. From the above reason the original full system is usually decomposed into two smaller subsystems, i. e., the boundary layer and the reduced ones, which are analyzed instead of the full system [1].

The design problem of the full system can be also decomposed into two low-order ones. For example, in the stabilizing problem of linear singularly perturbed systems, stabilizing feedback control laws for the reduced system and the boundary layer system are decided independently, after this procedure the composite control which stabilizes the original singularly perturbed system is constructed from these control laws [2],[3].

The object of the present analysis is (i) to show the controllability and the stabilizability of the reduced system (slow subsystem) are invariant under the fast feedback in nonlinear singularly perturbed systems, (ii) a composite control for the nonlinear singularly perturbed system is proposed and the property of this control is discussed.

Consider the system of the equation :

$$\begin{cases} \dot{x} = f(t, x, z, u) \\ \mu\dot{z} = g(t, x, z, u) \end{cases} \qquad (\cdot \overset{\Delta}{=} \frac{d}{dt}) \qquad \begin{matrix} x(t_o) = x_o \\ z(t_o) = z_o \end{matrix} \tag{1}$$

where x and z are n- and m-dimensional state vectors, respectively, u is an r-dimensional control vector and μ is a small positive parameter.

Assumption (i)

Functions $f, g,$ are continuously differentiable with respect to their all arguments.

Assumption (ii)

Mapping $g(t, x, \cdot, u) : R^m \to R^m$ is diffeomorphism for any fixed $t \geq 0, x \in R^n, u \in R^r$.

The system defined by (1) is called the full system. The reduced system of the full system is defined by setting $\mu = 0$ in (1),

$$\dot{x} = f(t, x, z, u) \tag{2-a}$$

$$0 = g(t, x, z, u) \ . \tag{2-b}$$

By the assumption (ii), (2-b) can be solved for z uniquely and denote this solution as

$$z = h(t, x, u) \ . \tag{3}$$

Then, z can be eliminated from the above equation (2) and the resultant equation gives the reduced system (slow subsystem)

$$\dot{x} = f(t, x, h(t, x, u), u) \ . \tag{4}$$

The boundary layer system is obtained from (1) after changing the time scale by introducing τ,

$$\mu\tau = (t - t_o)$$

where t_o is an initial time. Then, the boundary layer system (fast subsystem) is obtained in the form

$$\frac{dz}{d\tau} = g(t^*, x^*, z, u) , \tag{5}$$

where t^* is a scalar parameter and x^* is an n-dimensional vector parameter.

The reduced system (4) and the boundary layer system (5) play the key role in the analysis and the design problem of the singularly perturbed system (1). In the following we consider the feedback transformation such as,

$$u = \Gamma(t, z, v) .$$

In the first place this control acts as the feedback compensator for the fast subsystem (5). In the next place the control acts as a control for the reduced system (4) when the fast variable z in the control is eliminated. We finally determine v, such that the control becomes appropriate for the reduced system.

2. FEEDBACK INVARIANCE OF REDUCED SYSTEM CONTROLLABILITY AND STABILIZABILITY

In this section we consider the following feedback transformation in system (1) as

$$u = \Gamma(t, z, v) , \tag{6}$$

where $\Gamma(t, z, v)$ is an r-dimensional vector valued function and v is a new r-dimensional control vector and we assume that the mapping $\Gamma(t,z,\cdot)$: $R^r \rightarrow R^r$ is diffeomorphism for any fixed $t \geq 0$, $z \in R^m$. In this case, (6) can be solved uniquely for v as

$$v = L(t, z, u) \tag{6'}$$

Applying this feedback transformation to the full system (1), we have

$$\begin{cases} \dot{x} = f(t, x, z, \Gamma(t, z, v)) \\ \mu\dot{z} = g(t, x, z, \Gamma(t, z, v)) \end{cases} \tag{7}$$

The reduced system of (7) is

$$\dot{x} = f(t, x, z, \Gamma(t, z, v)) \tag{8-a}$$

$$0 = g(t, x, z, \Gamma(t, z, v)) \tag{8-b}$$

Again we assume that the equation (8-b) can be solved uniquely for z such as,

$$z = H(t, x, v) \ . \tag{9}$$

This gives the reduced system of (7) as,

$$\dot{x} = f(t, x, H(t, x, v), \Gamma(t, H(t, x, v), v)). \tag{10}$$

This system can be thought of as a modified reduced system, modification resulting from a feedback from the state variables of the fast subsystem.

Now we assume here that the reduced system (4) is controlled by $u = P(t)$. The dynamic behavior of the reduced system for this control is described by the following equation.

$$\dot{x} = f(t, x, h(t, x, P(t)), P(t)) \ , \quad x(t_o) = x_o \tag{11}$$

$$z = h(t, x, P(t)) \tag{12}$$

In the following lemma we claim that for any function $\Gamma(t,z,v)$, there exists $v(t)$ such that the dynamic behavior of the reduced system (10) coincides with (11).

Lemma : Assume that reduced system (2) and (8) can be solved uniquely for z respectively, then for any $P(t)$ and any $\Gamma(t, z, v)$, there exists $v(t)$ such that the dynamic behavior (10) completely coincides with the reduced system (11), i. e. the equation (10) becomes equivalent to (11).

proof : By comparing (2) and (8), and noting the existence and the uniqueness of differential equations it is clear that if the next identity holds

$$\Gamma(t, z, v(t)) \equiv P(t) \quad \text{(for any } t) \tag{13}$$

then, the system (2) and (8) become equivalent. But this directly means equivalence of (10) and (11) by the assumption. By using (6') and substituting (12) into (13) we can show that the above identity holds if we choose $v(t)$ as,

$$v(t) = L(t,\ h(t,\ x(t),\ P(t)),\ P(t)) \tag{14}$$

where $x(t)$ is the solution of (11). This completes the proof.

From the above discussion, we have the following theorem.

Theorem 1 : If (2-b) and (8-b) can be solved uniquely for z, the reduced system (10) is controllable if and only if the reduced system (4) is controllable. Moreover, the controllability of the reduced system (4) is invariant under the feedback transformation $u = \Gamma(t,z,v)$.

Proof : It is clear that the change in the control variable u,

$$u = \Gamma(t,\ z,\ v)$$

will transform the reduced system (4) into the reduced system (10). By the lemma we can conclude that for any control input in the system (4), the behavior of the system (10) coincides with the behavior of the system (4) whenever control $v(t)$ of (14) is used. Hence the controllability of (4) means that of (10) and vice versa, and the remaining part of the theorem is immediately follows.

This theorem is the generalized result of the theorem given by chow [4].

Next we consider the invariance of reduced system stabilizability. Assume that the reduced system (4) is stabilizable and $u = M(t,\ x)$ is stabilizing control for (4), i. e.,

$$\dot{x} = f(t,\ x,\ h(t,\ x,\ M(t,\ x)),\ M(t,\ x)) \tag{15}$$

$$z = h(t,\ x,\ M(t,\ x)) \tag{16}$$

is asymptotically stable. Using the same argument as preceeding lemma's we can find v as the function of x and t, which satisfies the next identity.

$$\Gamma(t,\ z,\ v) \equiv M(t,\ x) \qquad (\text{for any } t,\ x) \tag{17}$$

Substituting (16) into (17), we obtain the following equation.

$$v(t,\ x) = L(t,\ h(t,\ x,\ M(t,\ x)),\ M(t,\ x)) \tag{18}$$

Using this feedback control for (10), we will have the system (15), which is asymptotically stable. Thus we succeeded to stabilize the reduced system (10) and this gives the next theorem.

Theorem 2 : If (2-b) and (8-b) can be solved uniquely for z, then the reduced system (10) is stabilizable if and only if the reduced system (4) is stabilizable. And the stabilizability of the reduced system (4) is invariant under the feedback transformation $u = \Gamma(t, z, v)$. Moreover, a stabilizing control law for the modified reduced system (10) is given by (18).

3. A COMPOSITE CONTROL AND ITS PROPERTY

We define the composite control for the full system (1) as,

$$u = \Gamma(t, z, L(t, h(t, x, M(t, x)), M(t, x))) . \tag{19}$$

Applying this composite control for the full system (1), we have

$$\left\{ \begin{aligned} \dot{x} &= f(t, x, z, \Gamma(t, z, L(t, h(t, x, M(t, x)), M(t, x))) \\ \mu\dot{z} &= g(t, x, z, \Gamma(t, z, L(t, h(t, x, M(t, x)), M(t, x))) . \end{aligned} \right. \tag{20}$$

Since (19) is $u = \Gamma(t, z, v)$ and $v = L(t, h(t, x, M(t, x)), M(t, x))$ by the lemma it follows that the reduced system of the full system (20) is given by

$$\dot{x} = f(t, x, h(t, x, M(t, x)), M(t, x)), \tag{21}$$

which is the same equation as (4) with $u = M(t, x)$. The boundary layer system becomes

$$\frac{dz}{d\tau} = g(t^*, x^*, z, \Gamma(t^*, z, v^*)) , \tag{22}$$

where $t^*, x^*, v^* \overset{\Delta}{=} L(t^*, h(t^*, x^*, M(t^*, x^*)), M(t^*, x^*))$ are parameters which belong to $t^* \in R^1, x^* \in R^h, v^* \in R^r$ respectively. Equations (21) and (22) suggest us the possibility that the boundary layer stabilization can be performed without changing the dynamic behavior of the reduced system. It is of crucial importance as the property of the composite control that the implemantation of its control for the full system (1) permits us following two stage design procedure. In the first stage we design the feedback control $M(t, x)$ which compensates the dynamical property of the reduced system and in the second stage we choose the feedback compensator $\Gamma(t, z)$ for the boundary layer system. In the following we consider the stabilization problem of the linear singularly perturbed system as the example of this design procedure.

Example : The problem considered here is to stabilize the linear singularly perturbed system

$$\begin{cases} \dot{x} = A_1(t)x + A_2(t)z + B_1(t)u \\ \mu\dot{z} = A_3(t)x + A_4(t)z + B_2(t)u \end{cases} \tag{23}$$

where all the matrices $A_i(t)$, $B_i(t)$ and their derivatives are bounded and continuous function of t for all $t \geq t_o$, and $A_{22}(t)$ has an inverse for each fixed t. The reduced system of (23) is

$$\dot{x} = A_o(t)x + B_o(t)u \tag{24}$$

where $A_o(t) = A_1(t) - A_2(t)A_4(t)^{-1}A_3(t)$, $B_o(t) = B_1(t) - A_2(t)A_4^{-1}(t)B_2(t)$. The boundary layer system is given by

$$\frac{dz}{d\tau} = A_3(t^*)x^* + A_4(t^*)z + B_2(t^*)u \tag{25}$$

where t^* is a parameter.

Theorem 3 : If (24) is uniformly asymptotically stabilizable and $(A_4(t^*), B_2(t^*))$ is the controllable pair for each $t^* \geq 0$. Then the full system (23) is uniformly asymptotically stabilizable.

proof : By the assumption we can find $\hat{K}(t)$ such that the reduced system becomes uniformly asymptotically stable by using the feedback $u = \hat{K}(t)x$ and also we can find the feedback control $u = K(t)z$ for (25) such that the real parts of all the eigenvalues of $A_4(t^*) + B_2(t^*)K(t^*)$ are smaller than a fixed negative number for each $t^* \geq t_o$. Now we construct the composite control (19) for the system (23). In this case we can put $h(t, x, u)$, $\Gamma(t, z, v)$, $L(t, z, u)$, $M(t, x)$ as

$$h(t, x, u) = -A_4^{-1}(t)(A_3(t)x + B_2u)$$

$$\Gamma(t, z, v) = K(t)z + v$$

$$L(t, z, u) = u - K(t)z$$

$$M(t, x) = \hat{K}(t)x \quad .$$

Then, from (19), we can obtain the composite control as,

$$u = K(t)z + \{K(t)A_4^{-1}(t)(A_3(t) + B_2\hat{K}(t)) + \hat{K}(t)\}x \quad . \tag{26}$$

Using this control for (23), we obtain the reduced system and the boundary layer system from (21) and (22) as follows.

$$\dot{x} = A_o(t)x + B_o(t)\hat{K}(t)x \tag{27}$$

$$\frac{dz}{d\tau} = A_3(t^*)x^* + [A_4(t^*) + B_2(t^*)K(t^*)]z + B_2(t^*)v^* \tag{28}$$

But (27) is uniformly asymptotically stable and all the eigenvalues of $A_4(t^*) + B_2(t^*)K(t^*)$ are smaller than a fixed negative number for each $t^* \geq t_o$. This fact is sufficient to assure the uniformly asymptotically stable of the original singularly perturbed system controlled by (26), by the theorem in [5].

The composite control (26) is the stabilizing control for the singularly perturbed system (23). O'Reilly [9] obtained the same stabilizing control (26) by the another way. The composite control law in [2-3], [6-8] are in the same class of (26) but often they satisfy different criteria, optimality, etc.

4. CONCLUSION

It is shown that the controllability and the stabilizability of the reduced system of the singularly perturbed nonlinear system are invariant to the class of fast feedback controls. This result generalizes the earlier result in [4] from linear to nonlinear singularly perturbed systems. The general type composite control for the singularly perturbed nonlinear system is proposed and, as an example, its application for the stabilizing problem is considered.

REFERENCES

[1] P. V. Kokotovic, R. E. O'Malley, Jr. and P. Sannuti,"Singular perturbations and order reduction in control theory — An overeview," Automatica, 12, pp. 123-132, 1976.

[2] M. Suzuki and M. Miura,"Stabilizing feedback controllers for singulaly perturbed linear constant systems, IEEE Trans. Automat. Contr., Vol. AC-21, pp. 123-124, Feb. 1976.

[3] J. H. Chow and P. V. Kokotovic,"A decomposition of near-optimum regulators for systems with slow and fast modes," IEEE Trans. Automat. Contr., Vol. AC-21, pp. 701-705, Oct. 1976.

[4] J. H. Chow,"Preservation of controllability in linear time-invariant perturbed systems," Int. J. Contr., Vol. 25, No. 5, pp. 697-704, 1977.

[5] R. R. Wilde and P. V. Kokotovic,"Stability of singularly perturbed systems and networks with parasitice," IEEE Trans. Aut. Contr., AC-17, pp. 245-246, 1972.

[6] J. H. Chow, P. V. Kokotovic,"Two-time-scale feedback design of a class of nonlinear systems," IEEE Trans. Automat. Contr., Vol. AC-23, No. 3, 1978.

[7] J. H. Chow and P. V. Kokotovic,"Near-optimal feedback stabilization of a class of nonlinear singularly perturbed systems," SIAM J. Contr., Optimiz., Vol. 16, 1978.

[8] P. Sannuti,"On the controllability of singularly perturbed systems," IEEE Trans. Automat. Contr., Vol. AC-22, pp. 622-624, August 1977.

[9] J. O'Reilly,"Two time-scale feedback stabilization of linear time-varying singularly pertubed systems," L. Franklin Institute, Vol. 308, pp. 465-474, November 1979.

A TENSOR APPROACH TO MODELING OF NONHOMOGENEOUS NONLINEAR SYSTEMS*

STEPHEN YURKOVICH AND MICHAEL SAIN
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

ABSTRACT

Model following control methodology plays a key role in numerous application areas. Cases in point include flight control systems and gas turbine engine control systems. Typical uses of such a design strategy involve the determination of nonlinear models which generate requested control and response trajectories for various commands. Linear multivariable techniques provide trim about these motions; and protection logic is added to secure the hardware from excursions beyond the specification range. This paper reports upon experience in developing a general class of such nonlinear models based upon the idea of the algebraic tensor product.

INTRODUCTION

One of the most useful design procedures in control systems applications is that of model following control [1]. Current advanced applications in flight control [2] and in gas turbine engine control [3] directly involve such notions, applied in the global, nonlinear sense. Even in the case of trim, which is carried out with the aid of various versions of linear multivariable control theory, the model following idea has received much attention. See for example the classic papers of Morse [4] and of Wang and Davison [5]. More recently, the idea of linear model following has been applied simultaneously to control action and response variables by Peczkowski, Sain, and Leake [6] and by Sain and Ma [7], who formulated the problem in module terms. The ideas of [6] have been extended to a larger class of plants by Desoer and Chen [8].

In any type of model following design strategy, selection of the model whose behavior the plant is to emulate becomes a matter of the most essential importance. Many challenging issues remain along these lines, even in the linear multivariable case. When nonlinearities are to be included, these questions are multiplied many times over.

This paper reports upon a continuing investigation aimed at developing a general class of nonlinear differential models, which can be tailored to specific applications through a process of parameter selection or identification. The basis for the model is taken in the arena of series expansion; but the emphasis is placed upon the use of algebraic tensors to organize and to manipulate the details involved, which would otherwise become unmanageable.

POWER SERIES AND ALGEBRAIC TENSORS

Consider the nonlinear ordinary differential equation

$$\dot{x} = f(x,u) \ , \tag{1}$$

---

*This work was supported by the National Aeronautics and Space Administration under Grant NSG 3048.

where $x \in X$ and $u \in U$ for real vector spaces X and U of dimension n and m, respectively. Without loss, assume that $(0,0) \in X \times U$ is an equilibrium point of (1). The function $f : X \times U \to X$ may be required to satisfy certain technical conditions, as for example analyticity in a neighborhood of $(0,0)$ in $X \times U$. However, these conditions will not be stressed in the following sections. It is assumed, however, that trajectories initiated within a suitable neighborhood of the additive unit in the group X have that unit as their limit in time, when the choice $u(t) \equiv 0$ is made.

The right member of (1), under the technical assumptions just mentioned, can be expressed in the manner

$$f(x,u) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} \frac{1}{j!k!} M_{jk} \, (\underbrace{x \otimes x \otimes \cdots \otimes x}_{j \text{ times}} \otimes \underbrace{u \otimes u \otimes \cdots \otimes u}_{k \text{ times}}) , \qquad (2)$$

where $\otimes : X \times U \to X \otimes U$ is the universal bilinear tensor product function [9,10] and $X \otimes U$ is the nm-dimensional real vector space which is the tensor product of X with U. Elements of the form shown in (2) are built up by iteration, as for example $(X \otimes (X \otimes U)) \otimes U$. The main feature of the algebraic tensor representation (2) lies in the fact that the M_{jk} are linear maps. This allows for an orderly treatment of higher order terms in the expansion. For simplicity, write

$$L_{jk} = \frac{1}{j!k!} M_{jk} \qquad (3)$$

for the linear map whose action is that of M_{jk} followed by the indicated scalar multiplication. An introductory discussion of the idea of tensor product and its use to extract the linear part of a bilinear behavior may be found in [11].

A nonlinear model of (1) can then be obtained by truncation of (2).

INTERCONNECTIONS AND FEEDBACK CONTROL

Before illustrating the possibility for the use of such truncations in nonlinear model building, it is desirable to point out two features, which make such truncations especially interesting. The first of these features is the basic compatibility of series truncation with the notion of the nonlinear quotient signal flowgraph [12]. Intuitively, this means that truncation has the potential to be homomorphically related to series and parallel connections. The second of the features is the possibility for application of the models in feedback control, as studied by Buric [13].

APPLICATION TO THE MODELING PROBLEM

The discussion now turns to application of these notions in generation of nonlinear models for nonhomogeneous differential systems. Consider again a series expansion in the spirit of (2) for the n-vector x and the m-vector u. Since the task is to identify the parameters of the L_{jk} operators, the new equation

$$\dot{x} = [L_{10} \; L_{01} \; L_{20} \; L_{11} \; L_{02} \; L_{30} \; \ldots] \, x_T \qquad (4)$$

would be constructed, where x_T is a vector partitioned into tensor product terms in an appropriate order, given by $(x, u, x \otimes x, x \otimes u, u \otimes u, x \otimes x \otimes x, \ldots)$. Consider the tensor product associated with the L_{21} operator, whose 18 monomials can be represented conveniently as

$$x \otimes x \otimes u = \begin{bmatrix} x_1^2u_1 & x_1^2u_2 & x_1x_2u_1 & x_1x_2u_2 & x_1x_3u_1 & x_1x_3u_2 \\ x_2x_1u_1 & x_2x_1u_2 & x_2^2u_1 & x_2^2u_2 & x_2x_3u_1 & x_2x_3u_2 \\ x_3x_1u_1 & x_3x_1u_2 & x_3x_2u_1 & x_3x_2u_2 & x_3^2u_1 & x_3^2u_2 \end{bmatrix}, \tag{5}$$

when $n = 3$ and $m = 2$. Notice that there are six identical terms in the product so that elimination of such redundant terms leaves an array of dimension 12. A problem of reduced order is formulated by reducing the size of each tensor product term, and thus the size of the corresponding matrix of L_{jk}, which amounts to a use of the symmetric tensor algebra [10]. In the remainder of the discussion, we will continue to label these operators as L_{jk}, remembering that they now refer to the reduced order structure.

The number of distinct elements from each product is given by the combinatorial expression

$$p = \binom{n + q - 1}{q} \cdot \binom{m + r - 1}{r} \tag{6}$$

for q copies of x and r copies of u in the product. Construction of the system consists of stacking these monomials in the vector x_T to give a reduced-size version of (4).

To complete the construction of our approximate system, sinusoidal inputs are applied to (1), and the state solutions are sampled at h selected time points. These sampled values are loaded into the pxh matrix X_T. The first n+m rows of X_T are determined from the sampled values of x and u; the remaining $p-(n+m)$ rows contain monomials which are multiples of the entries of those first n+m rows. The nxh matrix $\dot{X}$ is formed by loading derivative estimates for $\dot{x}_1, \dot{x}_2, \ldots, \dot{x}_n$ at the h time points. As an illustration, for an approximation retaining up to third degree tensor product terms we have

$$\dot{X} = [L_{10}\ L_{01}\ L_{20}\ L_{11}\ L_{02}\ L_{30}\ L_{21}\ L_{12}\ L_{03}]\ X_T \,. \tag{7}$$

The method employed here in solving for the coefficient matrix uses a singular value decomposition of the transpose of X_T to solve the minimal least-squares problem, returning the nxp partitioned matrix of the L_{jk}.

After a suitable model is generated the task of model verification remains. The primary concern is, of course, to show that the simulated solution of the model matches the behavior of the true solution well in a region about the point of expansion. Moreover, a major feature demanded of the model is that it outperform the standard linearization given by the analytical expression for L_{10} and L_{01}, the usual "A" and "B" matrices of a linear approximation. Since there are an infinite number of possible initial states and amplitudes and frequencies of sinusoidal inputs in a region about the expansion point, some method of choosing an appropriate set of such combinations for the verification is necessary. The approach employed here, illustrated in the next section, is a "four-corner" scheme: that is, for the choice of the initial state x_0 we test at pairs

$(\bar{x}, \bar{x})$, $(\bar{x}, -\bar{x})$, $(-\bar{x}, -\bar{x})$, $(-\bar{x}, \bar{x})$ for several values of $\bar{x}$ near the point of expansion; thus, for an expansion about the origin we test at points along diagonals from the origin in the (x_1, x_2) plane. Similarly, for inputs

$$\begin{aligned} u_1 &= \alpha_1 \cos(2\pi\, \phi_1\, t) , \\ u_2 &= \alpha_2 \cos(2\pi\, \phi_2\, t) , \end{aligned} \tag{8}$$

we test at points $\alpha = (\alpha_1, \alpha_2)$ along diagonals in the four quadrants of the (α_1, α_2) plane. Finally, for combinations of such x_0 and α we test at various pairs of frequencies, $\phi = (\phi_1, \phi_2)$, for $\phi_1, \phi_2 \geq 0$.

Let us consider the method for which the comparison between the model simulation solution, x^M, and the linear approximation solution, x^L, is illustrated. Define ε_i^M as the absolute maximum error in x^M over the time range of simulation, T; that is,

$$\varepsilon_i^M \triangleq \max_{t\varepsilon T} |x_i^M(t) - x_i(t)| , \tag{9a}$$

where $x_i(t)$ is the true solution at time t, $i = 1,2,\ldots,n$. Similarly, define

$$\varepsilon_i^L \triangleq \max_{t\varepsilon T} |x_i^L(t) - x_i(t)| \tag{9b}$$

as the absolute maximum error in x^L over T. With these values, form the error ε_i which is the comparison $\varepsilon_i^M - \varepsilon_i^L$. Thus, if $\varepsilon_i \leq 0$, the model has performed at least as well as the linear approximation. Obviously, the ideal is to have $\varepsilon_i < 0$ for all simulations. An error analysis using measures (9a,b) will be used to illustrate the performance of the models constructed in the following example.

EXAMPLE

The computational aspects of the modeling scheme will now be discussed by way of example. Let the state x be a 2-vector (x_1,x_2) and the input u a 2-vector (u_1,u_2). Consider the system

$$\begin{aligned} \dot{x}_1 &= u_2 \cosh(x_1x_2) - e^{2u_1} \sinh(2x_1) - 3\sinh(x_2) , \\ \dot{x}_2 &= e^{u_1u_2} \sinh(x_1) - e^{u_1} u_1 \cosh(x_1^2) + \sinh(x_2) , \end{aligned} \tag{10}$$

where u is given by (8). Since $\dot{x}(0,0)$ is zero, the origin is an equilibrium point, and will thus be the point of expansion. A quick calculation reveals that the operators which form the linear approximation are given by

$$L_{10} = \begin{bmatrix} -2 & -3 \\ 1 & 1 \end{bmatrix} , \qquad L_{01} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} . \tag{11}$$

The eigenvalues of L_{10} have negative real parts, verifying that the origin is a stable equilibrium point. In order to formulate (7) for the least squares minimal solution, (10) is integrated and the resulting solution sampled over a four second interval; 200 samples, evenly spaced at 0.02 seconds, are taken. In the identification the system is perturbed by choosing the initial state (0.005,-0.005), the input amplitudes (0.05,0.05), and the frequencies (0.75, 1.0) for the sinusoidal inputs (all frequencies in hertz).

An approximation keeping third degree tensor product terms results in a model with nine linear operators:

$$L_{10} = \begin{bmatrix} -2.003 & -3.007 \\ 1.012 & 1.013 \end{bmatrix}, \; L_{01} = \begin{bmatrix} -0.011 & 1.008 \\ -0.992 & -0.005 \end{bmatrix},$$

$$L_{20} = \begin{bmatrix} 0.534 & 2.406 & 0.918 \\ -0.532 & -0.703 & 0.078 \end{bmatrix}, \; L_{11} = \begin{bmatrix} -3.223 & -0.100 & -0.415 & 0.471 \\ -0.565 & 0.202 & 0.086 & -0.175 \end{bmatrix},$$

$$L_{02} = \begin{bmatrix} 0.254 & -0.223 & 0.027 \\ -1.174 & 0.155 & -0.023 \end{bmatrix}, \; L_{30} = \begin{bmatrix} -29.75 & -38.17 & 110.5 & 77.99 \\ 5.877 & -23.09 & -77.32 & -29.40 \end{bmatrix},$$

$$L_{21} = \begin{bmatrix} -11.15 & 0.430 & 25.74 & -12.66 & 24.31 & -3.399 \\ 7.179 & -0.973 & -23.63 & 17.73 & -18.47 & 8.554 \end{bmatrix},$$

$$L_{12} = \begin{bmatrix} -11.56 & 6.837 & -0.975 & 7.570 & -9.546 & 3.216 \\ 7.109 & -2.496 & 0.655 & -4.076 & 7.813 & -2.156 \end{bmatrix},$$

$$L_{03} = \begin{bmatrix} -0.205 & -0.158 & 0.087 & 0.037 \\ -0.074 & 0.171 & -0.206 & -0.033 \end{bmatrix}. \qquad (12)$$

Note that L_{10} and L_{01} approximate (11) closely. Simulation of (12) for various initial conditions and input amplitudes at a fixed frequency pair $\phi = (0.75, 1.0)$ indicates that the model outperforms the linear approximation of (11) in a region about the equilibrium point. Samples of these tests are represented in Table 1, all calculations done in single precision arithmetic. Figure 1 illustrates the behavior of x_1 for $\alpha = (0.25, -0.25)$ at $x_0 = (-0.01, 0.01)$; while the model solution outperforms the linear approximation, it may be deemed unsatisfactory for some applications. Results for simulations of this model at various frequency pairs are given in Table 2, where it is seen that the linear solution outperforms the model solution for most choices of α.

These last two difficulties sighted in the degree-three approximation model of (12) can be cured by employing a second degree approximation to construct the model. Under the identical conditions x_0, α, and ϕ used in the identification of (12), the five individual linear operators comprising this degree-two model are:

$$L_{10} = \begin{bmatrix} -2.001 & -3.009 \\ 1.006 & 1.011 \end{bmatrix}, \quad L_{01} = \begin{bmatrix} 0.002 & 0.997 \\ -1.000 & 0.000 \end{bmatrix},$$

$$L_{20} = \begin{bmatrix} 0.239 & -0.145 & -0.720 \\ -0.323 & -0.128 & 0.359 \end{bmatrix}, \quad L_{02} = \begin{bmatrix} -0.105 & 0.027 & 0.012 \\ -0.982 & 0.015 & -0.013 \end{bmatrix},$$

$$L_{11} = \begin{bmatrix} -4.15 & -0.074 & -0.048 & -0.176 \\ -0.007 & 0.083 & 0.008 & 0.102 \end{bmatrix}. \tag{13}$$

As a comparison, Figure 2 offers a simulation of (13) for the same conditions as those in the simulation of the degree-three model of Figure 1; considerable improvement is evident. Several simulations of (13) are given in Tables 3 and 4, allowing a comparison with Tables 1 and 2. Again, marked improvement is noticed with the degree-two model, particularly for the frequency variation analysis in Table 4. An interesting feature of this example concerns the sensitive behavior of (10) for low frequency inputs; in the D.C. case, input amplitude steps of over 0.1 in magnitude cause instability in the system. The next two figures serve to illustrate low frequency behavior of the system and the ability of the model to track. Figure 3 depicts the response of x_2 for $x_0 = (0.01, -0.01)$, $\alpha = (-0.075, -0.075)$, and $\phi = (0.05, 0.05)$. A step response (i.e., $\phi = (0.0, 0.0)$) is given in Figure 4 for x_1 at $x_0 = (0.01, -0.01)$ and $\alpha = (-0.075, -0.075)$. In both instances the model solution matches the true solution well, while the linear approximation is poor.

We have seen that the equation of this example in some sense behaves as a 2nd-degree system. The model obtained via a third degree approximation was offered to illustrate the extension, as the added computational complexity in the identification is minimal. There are, of course, systems which would demand the use of the higher degree approximation.

ACKNOWLEDGEMENTS

The authors wish to express their appreciation to T. Klingler for his assistance in producing the numerous error analyses in the model verification.

REFERENCES

1. E. Kreindler and D. Rothchild, "Model Following in Linear Quadratic Optimization", Journal of the American Institute of Aeronautics and Astronautics, Volume 14, pp. 835-842, July 1976.

2. G. Meyer, "Use of System Inverses in the Design of Flight Control Systems for Aircraft of Nonlinear Characteristics", Joint Automatic Control Conference, San Francisco, California, August 13-15, 1980.

3. R.L. DeHoff, W. Earl Hall, Jr., R.J. Adams, and N.K. Gupta, "F100 Multivariable Control Synthesis Program", Volumes I and II, Technical Report AFAPL-TR-77-35, June 1977.

4. A.S. Morse, "Structure and Design of Linear Model Following Systems", IEEE Transactions on Automatic Control, Volume 18, pp. 346-353, August 1973.

5. S.-H. Wang and E.J. Davison, "A Minimization Algorithm for the Design of Linear Multivariable Systems", IEEE Transactions on Automatic Control, Volume 18, pp. 220-225, June 1973.

6. J.L. Peczkowski, M.K. Sain, and R.J. Leake, "Multivariable Synthesis with Inverses", Proceedings 1979 Joint Automatic Control Conference, pp. 375-380.

7. M.K. Sain and A. Ma, "Multivariable Synthesis with Reduced Comparison Sensitivity", Proceedings 1980 Joint Automatic Control Conference, Part WP8-B.

8. C.A. Desoer and M.J. Chen, "Algebraic Design of Unity-Feedback Systems with Stable Plant", Proceedings 1980 Joint Automatic Control Conference, Part TA8-E.

9. S. MacLane and G. Birkhoff, Algebra. London: Macmillan, 1967, Chapters 9 and 16.

10. W.H. Greub, Multilinear Algebra. New York: Springer-Verlag, 1967.

11. M.K. Sain, "The Growing Algebraic Presence in Systems Engineering: An Introduction", Proceedings of the IEEE, Volume 64, No. 1, pp. 96-111, January 1976.

12. M.K. Sain, "Quotient Signal Flowgraphs: New Insights", Proceedings IEEE International Conference on Circuits and Computers, p. 417, October 1980.

13. M.R. Buric, "Application of Multilinear Algebra in Optimal Regulation of Nonlinear Polynomial Systems", Proceedings Sixteenth Allerton Conference on Communication, Control, and Computing, pp. 655-663, 1978.

Table 1: Error Analyses for Degree-3 Model; ϕ = (0.75, 1.0)

($\times 10^{-2}$)		($\times 10^{-1}$)		($\times 10^{-1}$)	
x_{10}	x_{20}	α_1	α_2	ε_1	ε_2
0.1	0.1	0.0	0.0	0.000	0.000
0.1	0.1	0.5	0.5	-0.042	-0.026
0.1	0.1	0.5	-0.5	-0.040	-0.024
0.1	0.1	-0.5	-0.5	-0.044	-0.026
0.1	0.1	-0.5	0.5	-0.039	-0.025
0.1	0.1	2.5	2.5	-1.080	-0.709
0.1	0.1	2.5	-2.5	-0.851	-0.627
0.1	0.1	-2.5	-2.5	-0.991	-0.698
0.1	0.1	-2.5	2.5	-1.130	-0.664
-0.5	-0.5	0.0	0.0	0.000	0.000
-0.5	-0.5	0.5	0.5	-0.043	-0.028
-0.5	-0.5	0.5	-0.5	-0.040	-0.025
-0.5	-0.5	-0.5	-0.5	-0.051	-0.028
-0.5	-0.5	-0.5	0.5	-0.041	-0.025
-0.5	-0.5	2.5	2.5	-1.080	-0.719
-0.5	-0.5	2.5	-2.5	-0.845	-0.625
-0.5	-0.5	-2.5	-2.5	-1.030	-0.722
-0.5	-0.5	-2.5	2.5	-1.170	-0.665
-1.0	1.0	0.0	0.0	0.003	0.002
-1.0	1.0	0.5	0.5	-0.042	-0.025
-1.0	1.0	0.5	-0.5	-0.041	-0.024
-1.0	1.0	-0.5	-0.5	-0.040	-0.024
-1.0	1.0	-0.5	0.5	-0.040	-0.025
-1.0	1.0	2.5	2.5	-1.070	-0.705
-1.0	1.0	2.5	-2.5	-0.855	-0.626
-1.0	1.0	-2.5	-2.5	-0.990	-0.700
-1.0	1.0	-2.5	2.5	-1.120	-0.666

Table 2: Error Analyses for Degree-3 Model; x_0 = (0.01, -0.01)

($\times 10^{-2}$)		($\times 10^{-2}$)		($\times 10^{-1}$)	
α_1	α_2	ϕ_1	ϕ_2	ε_1	ε_2
1.0	-7.5	0.0	0.0	0.476	0.372
1.0	-7.5	1.0	1.0	0.461	0.360
1.0	-7.5	2.5	2.5	0.389	0.306
1.0	-7.5	5.0	5.0	0.378	0.295
-5.0	5.0	0.0	0.0	1.400	0.934
-5.0	5.0	1.0	1.0	1.380	0.928
-5.0	5.0	2.5	2.5	1.310	0.896
-5.0	5.0	5.0	5.0	1.060	0.782
-7.5	1.0	0.0	0.0	1.160	-0.631
-7.5	1.0	1.0	1.0	1.150	-0.614
-7.5	1.0	2.5	2.5	1.090	0.596
-7.5	1.0	5.0	5.0	0.977	0.659
-7.5	-7.5	0.0	0.0	-0.550	-0.528
-7.5	-7.5	1.0	1.0	-0.542	-0.512
-7.5	-7.5	2.5	2.5	-0.477	-0.423
-7.5	-7.5	5.0	5.0	-0.437	-0.404

Table 3: Error Analyses for Degree-2 Model; ϕ = (0.75, 1.0)

($\times 10^{-2}$)		($\times 10^{-1}$)		($\times 10^{-1}$)	
x_{10}	x_{20}	α_1	α_2	ε_1	ε_2
0.1	0.1	0.0	0.0	0.000	0.000
0.1	0.1	0.5	0.5	-0.044	-0.027
0.1	0.1	0.5	-0.5	-0.042	-0.025
0.1	0.1	-0.5	-0.5	-0.045	-0.027
0.1	0.1	-0.5	0.5	-0.042	-0.025
0.1	0.1	2.5	2.5	-1.270	-0.720
0.1	0.1	2.5	-2.5	-1.320	-0.635
0.1	0.1	-2.5	-2.5	-1.370	-0.717
0.1	0.1	-2.5	2.5	-1.121	-0.656
-0.5	-0.5	0.0	0.0	0.000	0.000
-0.5	-0.5	0.5	0.5	-0.043	-0.028
-0.5	-0.5	0.5	-0.5	-0.043	-0.026
-0.5	-0.5	-0.5	-0.5	-0.050	-0.028
-0.5	-0.5	-0.5	0.5	-0.041	-0.026
-0.5	-0.5	2.5	2.5	-1.131	-0.715
-0.5	-0.5	2.5	-2.5	-1.320	-0.635
-0.5	-0.5	-2.5	-2.5	-1.410	-0.721
-0.5	-0.5	-2.5	2.5	-1.200	-0.654
-1.0	1.0	0.0	0.0	0.001	0.001
-1.0	1.0	0.5	0.5	-0.043	-0.026
-1.0	1.0	0.5	-0.5	-0.042	-0.025
-1.0	1.0	-0.5	-0.5	-0.043	-0.025
-1.0	1.0	-0.5	0.5	-0.041	-0.025
-1.0	1.0	2.5	2.5	-1.250	-0.072
-1.0	1.0	2.5	-2.5	-1.330	-0.635
-1.0	1.0	-2.5	-2.5	-1.340	-0.716
-1.0	1.0	-2.5	2.5	-1.210	-0.656

Table 4: Error Analyses for Degree-2 Model; x_0 = (0.01, -0.01)

($\times 10^{-2}$)		($\times 10^{-2}$)		($\times 10^{-1}$)	
α_1	α_2	ϕ_1	ϕ_2	ε_1	ε_2
1.0	-7.5	0.0	0.0	-0.079	-0.079
1.0	-7.5	1.0	1.0	-0.080	-0.074
1.0	-7.5	2.5	2.5	-0.061	-0.061
1.0	-7.5	5.0	5.0	-0.062	-0.061
-5.0	5.0	0.0	0.0	-0.297	-0.266
-5.0	5.0	1.0	1.0	-0.492	-0.512
-5.0	5.0	2.5	2.5	-0.724	-0.673
-5.0	5.0	5.0	5.0	-0.745	-0.629
-7.5	1.0	0.0	0.0	-0.595	-0.525
-7.5	1.0	1.0	1.0	-1.010	-1.040
-7.5	1.0	2.5	2.5	-1.490	-1.370
-7.5	1.0	5.0	5.0	-1.430	-1.210
-7.5	-7.5	0.0	0.0	-0.528	-0.454
-7.5	-7.5	1.0	1.0	-0.704	-0.737
-7.5	-7.5	2.5	2.5	-1.080	-0.960
-7.5	-7.5	5.0	5.0	-1.010	-0.876

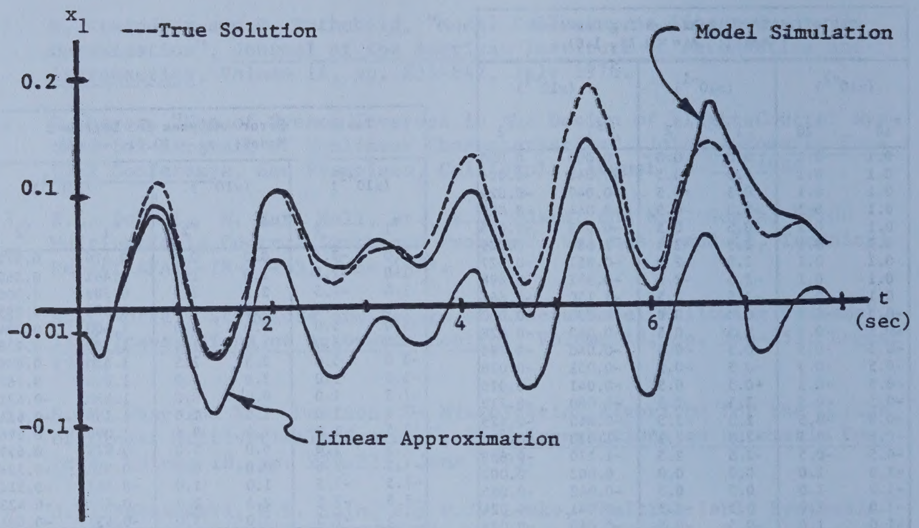


Figure 1 State x_1: Degree-3 Model

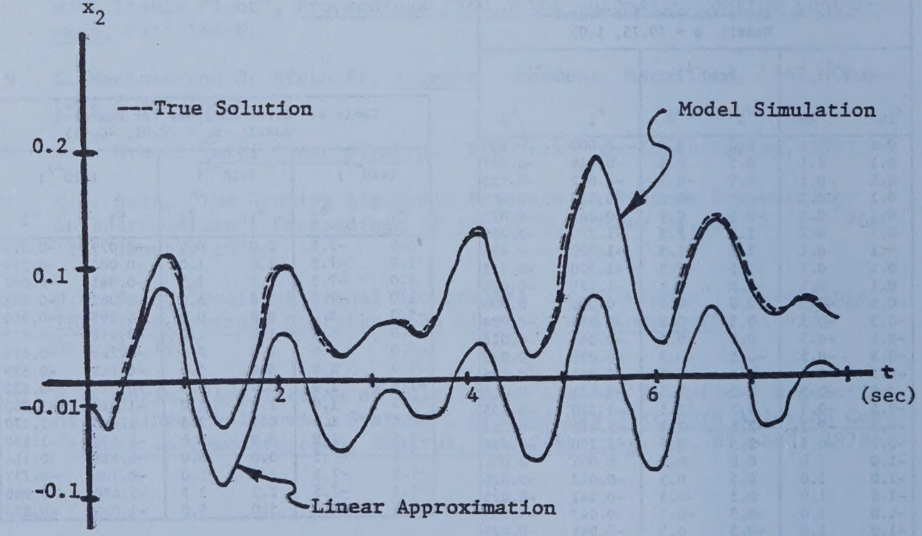


Figure 2 State x_1: Degree-2 Model

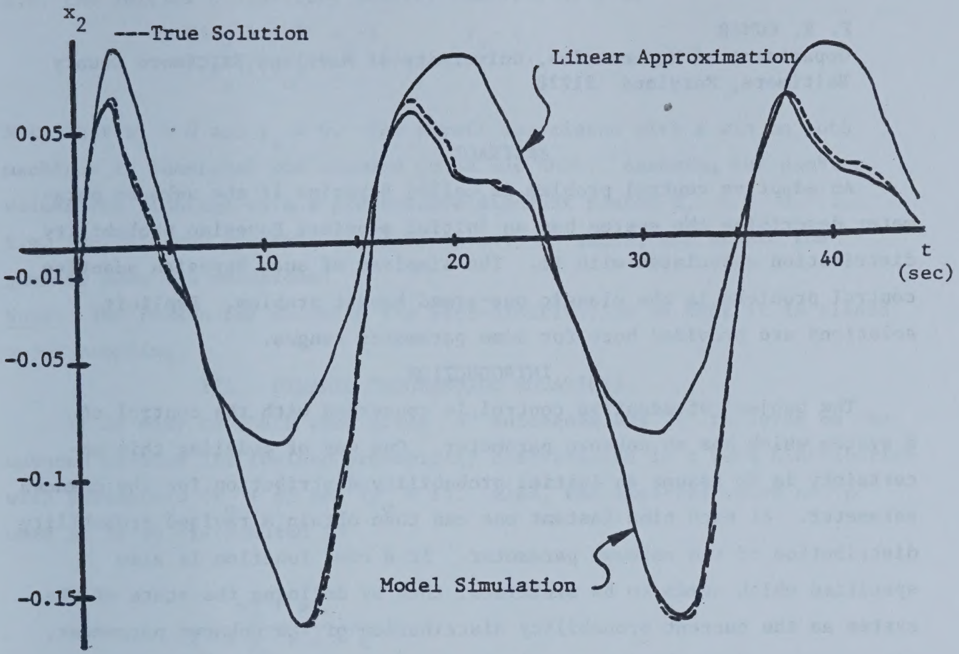


Figure 3 State x_2: $\phi = (0.05,0.05)$

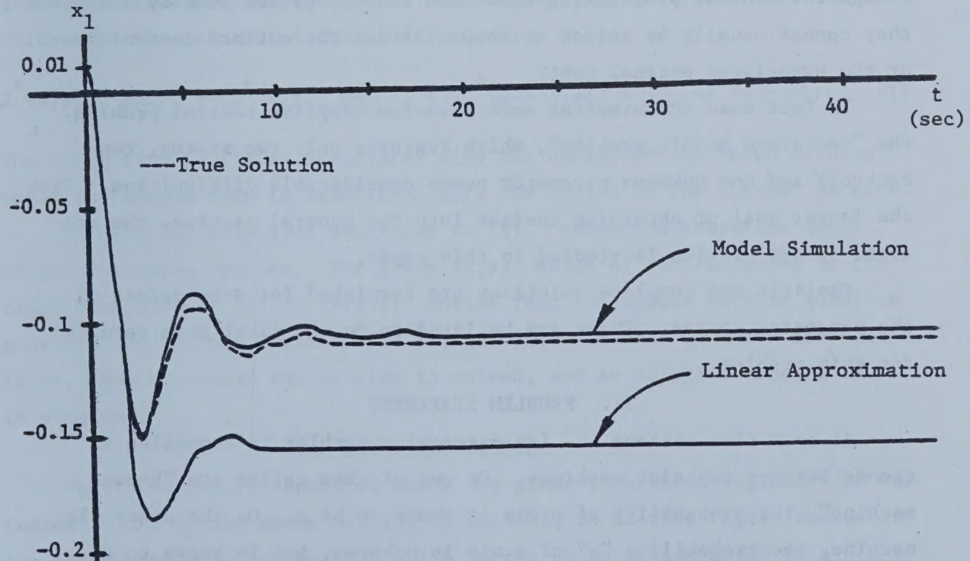


Figure 4 State x_1: $\phi = (0,0)$

THE SIMPLEST BAYESIAN ADAPTIVE CONTROL PROBLEM

P. R. KUMAR
Department of Mathematics, University of Maryland Baltimore County
Baltimore, Maryland 21228

ABSTRACT

An adaptive control problem is called Bayesian if the unknown parameter describing the system has an initial a-priori Bayesian probability distribution associated with it. The simplest of such Bayesian adaptive control problems is the classic one-armed bandit problem. Explicit solutions are provided here for some parameter ranges.

I. INTRODUCTION

The subject of adaptive control is concerned with the control of a system which has an unknown parameter. One way of modeling this uncertainty is to assume an initial probability distribution for the unknown parameter. At each time instant one can then obtain a revised probability distribution of the unknown parameter. If a cost function is also specified which needs to be minimized, then by defining the state of the system as the current probability distribution of the unknown parameter, one can proceed to obtain an optimal adaptive control law by solving the associated dynamic programming equations.

In practice a significant drawback of this procedure is that even though the dynamic programming equations can be written down by inspection, they cannot usually be solved to obtain either the optimal control law or the associated optimal cost.

In fact even the simplest such Bayesian adaptive control problem, the "one-armed bandit problem", which features only two states, two controls and one unknown parameter poses considerable difficulties. With the larger goal of obtaining insight into the general problem, the one-armed bandit problem is studied in this paper.

Explicit and complete solutions are furnished for some values of the parameter ranges. These are believed to be the first such results for this problem.

II. PROBLEM STATEMENT

At each time instant t (an integer) a gambler is compelled to choose between two slot machines. On one of them called the "known machine", the probability of a win is known to be q. On the other slot machine, the probability "p" of a win is unknown, but is known to be

---

The research reported here has been supported by the U.S. Army Research Office under Contract No. DAAG 29-80-K0038.

distributed according to the Beta-distribution with parameters x_o and y_o, i.e. the initial probability density function of p is

$$\frac{\Gamma(x_o + y_o)}{\Gamma(x_o)\Gamma(y_o)} p^{x_o - 1} (1 - p)^{y_o - 1} .$$

Note that $x_o > 0$ and $y_o > 0$. The payoff associated with a win on both machines is identical and assumed to be one unit. Assuming the gambler values his winnings with a progressive discount factor α, $0 < \alpha < 1$, i.e. one unit won at time t is worth only α^t units, how should the gambler make his decisions?

Note: The reason for choosing the Beta-distribution is that it is closed under sampling.

III. DYNAMIC PROGRAMMING EQUATIONS

It is easy to check that after "s" successes and "f" failures on the unknown machine its revised probability distribution is a Beta distribution with parameters $(x_o + s)$ and $(y_o + f)$. Also, the expected value of "p" when it is so distributed is

$$\frac{x_o + s}{x_o + y_o + s + f} .$$

Now define (x,y), the parameters of the Beta-distribution, as the state of system. Aslo let $J^*(x,y)$ be the optimal expected cost when starting at such a state. By using dynamic programming [1], one can immediately write down the associated functional equation as,

$$J^*(x,y) = \max[q + \alpha J^*(x,y); \frac{x}{x+y}[1 + \alpha J^*(x + 1,y)] + \frac{y}{x+y} \alpha J^*(x,y+1)] . \quad (1)$$

The first term in (1) is associated with the choice of the known machine while the second term is associated with the choice of the unknown machine.

No explicit solutions are known to (1). Attention therefore turns to the following problem. For given (x,y), which of the two terms on the right-hand side of (1) is larger? Notice that the answer to this question provides the optimal decision for the gambler. If this is known for each (x,y), then of course the problem is solved, and an adaptive control law is obtained.

IV. THE MAIN RESULTS

Our first result separates the x - y plane into three regions. In one region choice of the known machine is optimal, in another region choice of the unknown machine is optimal, while in the third region which is a "narrow" strip between the other two regions the optimal choice is unknown to us.

Theorem 1.

Let A and B be the straight lines.

$$A : x = \frac{q}{1-q}\, y - \frac{\alpha q}{(1-\alpha)(1-\alpha+\alpha q)}$$

$$B : x = \frac{q}{1-q}\, y .$$

In the region above A, choice of the unknown machine is optimal, while in the region below B choice of the known machine is optimal.

Proof: The proof is lengthy. It is given in [2]. □

Utilizing the above theorem, we have been able to come up with explicit solutions for some values of the parameter ranges.

Theorem 2.

Suppose $q = \frac{m}{m+n}$ where m and n are relatively prime positive integers. Also suppose that x_o and y_o are positive integers. Then for α such that $0 < \alpha \leq \frac{1}{n+1}$, the optimal solution to the problem is as follows:

$$\frac{x}{x+y} \geq q: \text{ choose the unknown machine}$$

$$\frac{x}{x+y} < q: \text{ choose the known machine.}$$

Proof: See [2]. □

Note that $\frac{x}{x+y}$ = E[Success on the unknown machine].
Hence this result is a "certainty-equivalent separation theorem".

V. CONCLUDING REMARKS

Optimal solutions to the one-armed bandit problem have been obtained for some regions in the state space. For some parameter values, complete explicit solutions have been obtained.

Since the one-armed bandit problem is the simplest Bayesian adaptive control problem, some of the techniques used in [2] may possibly be useful in more general problems. The obtaining of such solutions to more general problems will however be difficult, given the level of difficulty manifested by even this simple problem.

References to some other works on the one-armed bandit problem may be found in [2].

VI. REFERENCES

[1] R. BELLMAN, "A problem in the sequential design of experiments", Sankhya, Parts 3 and 4, Vol. 16, pp. 221-229, 1956.

[2] P. R. KUMAR and T. I. SEIDMAN, "On the optimal solution of the one-armed bandit adaptive control problem", Math. Research Report No. 80-14, University of Maryland Baltimore County, May 1980.

OPTIMAL ADAPTIVE CONTROLLERS FOR FINITE STATE SYSTEMS

P. R. KUMAR and A. BECKER
Department of Mathematics, University of Maryland Baltimore County
Baltimore, Maryland 21228

ABSTRACT

A new family of adaptive controllers is introduced for systems with finite state-spaces and finite control spaces. Without knowing what the model of the system is (or equivalently, without knowing what the parameter values describing the system are), the adaptive controllers achieve a long-term average cost precisely equal to the optimal long-term average cost attainable if the system were known. The adaptive controllers are therefore truly optimal.

I. INTRODUCTION

The design of good (optimal) adaptive controllers for unknown or incompletely known systems has been a long standing problem in automatic control ever since the design by James Watt of a governor for his steam engine.

Though early controllers were analog devices, in the past few decades increasing use has been made of digital computers in the feedback control of systems. A characteristic feature of such use is that the digital computer can classify an observed state of the system (through the use of an A/D converter for example) as one of a finite set of quantized states. Also, after the performance of some calculations, the digital computer can apply an input to the system which is chosen from a finite set of possible inputs. As far as the use of the digital computer in the feedback loop is concerned, the system therefore is effectively a finite-state, finite-control system.

In this paper we propose a new family of adaptive controllers for such finite-state, finite-control systems. The adaptive controller is unaware of the true model of the system, but is provided with a finite set of possible models one of which is the true one. Let us measure the performance of the system by the average cost incurred over a "long" operating time period. Each adaptive controller in our proposed family possesses the property that the resulting long-term average cost incurred is precisely equal to the optimal long-term average cost attainable if the true model were known. Hence the adaptive controllers are truly optimal.

The implementation of these adaptive controllers is particularly simple on any digital computer. Moreover, a designer can choose any one of

---

The research reported here has been supported by the U.S. Army Research Office under Contract No. DAAG 29-80-K0038.

the proposed infinite family of adaptive controllers, since all of them achieve optimal performance. Hence there exists considerable design flexibility.

Since these adaptive controllers have been specifically designed for digital-computer control of unknown or incompletely known systems, there exist a vast number of potential applications.

II. PROBLEM DESCRIPTION

Let X be the state-space of the system and U the set of controls available for application at each time instant. Let A be a set of possible models (or equivalently, parameter values) to describe the system. For each possible model $\alpha \in A$, let $p(i,j;\, u,\alpha)$ be the probability that the system will go from state $i \in X$ to state $j \in X$ under the application of $u \in U$.

Whenever the system undergoes such a transition, the cost incurred is $c(i,j,u)$.

Our goal is to minimize the long-term average cost,

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} c(x_s,\, x_{s+1},\, u_s)$$

where x_s is the state of the system at time s and u_s is the control input applied at time s. The true model for the system is not known, but it is known that it is one of the models in A.

Our only assumptions are the following.

i) X, U and A are finite sets.

ii) For each $(i,j) \in X \times X$,

$$\begin{array}{lll} \underline{\text{either}} & p(i,j;u,\alpha) > 0 & \text{for all } (u,\alpha) \in U \times A \\ \underline{\text{or}} & p(i,j;\, u,\alpha) = 0 & \text{for all } (u,\alpha) \in U \times A. \end{array} \qquad (1)$$

iii) For each $(i,j) \in X \times X$, there exist (a) an integer m_{ij}, and (b) a sequence of states $i = k_o, k_1, k_2, \ldots, k_{m_{ij}} = j$ such that

$$p(k_s,\, k_{s+1};\, u,\alpha) > 0 \quad \text{for } s = 0, 1, 2, \ldots, m_{ij} - 1.$$

Assumption (1.ii) requires the finite time probability measures associated with the various models in A to be mutually absolutely continuous. Since it is possible to eliminate in finite time, models which are not mutually absolutely continuous with respect to the true model, this is not a restrictive assumption. Assumption (1.iii) merely states that the system will wander through all states. Such an assumption is required for two reasons. Firstly, we want to "identify" the dynamics of the system, i.e. the transition probabilities for all possible transitions. Secondly, an assumption of this type, or more generally a single-ergodic class assumption is common in the theory of the long-term average cost criterion.

III. DESCRIPTION OF THE ADAPTIVE CONTROLLERS

For each possible model $\alpha \in A$, let $\phi(\cdot,\alpha): X \to U$ be an optimal feedback control law for model α, i.e. one which minimizes the long-term average cost for model α. Also let $J^*(\alpha)$ be the optimal such cost attainable for model α.

Choose an integer m and two functions f and o mapping the real line into itself satisfying:

i) $m > \max \{m_{ij} : i,j \in X\}$

ii) f is strictly monotone increasing with

$$f(J^*(\alpha)) > 0 \qquad \text{for all } \alpha \in A. \tag{2}$$

iii) $\lim\limits_{t\to\infty} o(t) = \infty$ while $\lim\limits_{t\to\infty} \frac{o(t)}{t} = 0.$

Construct the adaptive controllers as follows:

(a) At each time $t = 0, m, 2m, 3m, \ldots$ select $\hat{\alpha}_t$ so that it maximizes,

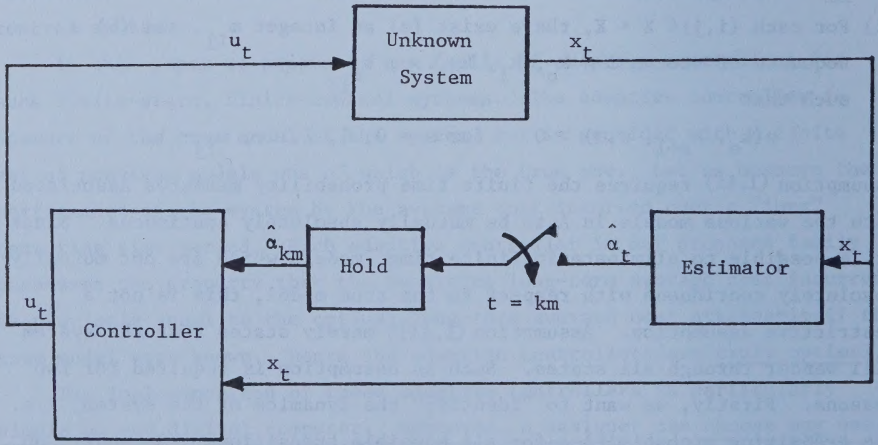
$$D_t(\alpha) := \left[\frac{1}{f(J^*(\alpha))}\right]^{o(t)} \prod_{s=0}^{t-1} p(x_s, x_{s+1}; u_s, \alpha) \tag{3}$$

over all $\alpha \in A$. Between km and $(k + 1)m$, keep $\hat{\alpha}_t$ constant, i.e. $\hat{\alpha}_t = \hat{\alpha}_{km}$ for $km \leq t < (k + 1)m$.

(b) At each time t, apply the control input,

$$u_t = \phi(x_t, \hat{\alpha}_t)$$

where x_t is the current state of the system.



IV. THE MAIN RESULTS

Our main results concerning the performance of this scheme are the following.

Theorem 1.

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} c(x_s, x_{s+1}, u_s) = J^*(\alpha^o) \text{ almost surely, where } \alpha^o \text{ is the}$$

true but unknown parameter.

Proof: The proof is lengthy and involves results from the theory of stochastic processes and the theory of optimal control with respect to the long-term average cost criterion. It is given in [1]. □

To describe the next result we need some additional notation. Let Γ be the set of all feedback control laws which are optimal for α^o. Clearly $\phi(\cdot, \alpha^o) \in \Gamma$, but there may be others. Let $\Gamma(i) := \{u \in U;\ \psi(i) = u$ for some $\psi \in \Gamma\}$ be the set of all controls u which are optimal inputs when the system is in state $i \in X$.

Theorem 2.

$$\lim_{t\to\infty} \frac{1}{t} \sum_{s=0}^{t-1} 1(u_t \in \Gamma(x_t)) = 1 \qquad \text{almost surely.}$$

Proof: See [1]. □

This result asserts that except for a "sparse" collection of times, or equivalently with "time-density one", an optimal control input is applied.

Note that we are not asserting that,

$$u_t \to \Gamma(x_t). \tag{4}$$

A counterexample to (4) is found in [1].

V. CONCLUDING REMARKS

Our main contribution is the introduction of the term $\left[\frac{1}{f(J^*(\alpha))}\right]^{o(t)}$ in (3). This term leads to the important results of Theorems 1 and 2. Without it, the adaptive control scheme is merely the well known method consiting of making a maximum-likelihood estimate of the unknown parameter at each time instant, and then applying a control input which is optimal for that estimate. Such a scheme works under restrictive conditions, namely an "identifiability condition" introduced by Mandl [2]. In general however, such a condition is too restrictive and as shown by Borkar and Varaiya [3] and Kumar [4], the straightforward maximum-likelihood method can lead to bad results. The introduction of the additional term

$\left[\frac{1}{f(J^*(\alpha))}\right]^{o(t)}$ and the sampling period m cures these difficulties. Recently Doshi and Shreve [5] have suggested another method, but this involves the use of randomized inputs, which may be unacceptable in practice.

An interesting feature of our method is that the optimal cost is always achieved, as specified in Theorem 1. Moreover "almost all the time", an optimal control input is applied to the system. However, we do not assert or require (in fact we provide a counterexample to) convergence of the parameter estimates or convergence of the adaptive controller to an optimal feedback control law.

Regarding their practical use, the optimal adaptive controllers suggested here are particularly easy to implement in any control scheme using a digital computer in the feedback loop. Also m, f and o can be any choices satisfying (2). Hence the design has considerable flexibility built into it.

In [1] simulation results, corraborating the theory, are presented of the performance of the optimal adaptive controller on an example which is especially hard to identify and control. In fact the straightforward maximum-likelihood method of [2, 3, 4] fails badly, however the optimal controller presented here, as expected, behaves optimally.

It is hoped to report on more substantive applications utilizing this scheme in the future. However, since these optimal adaptive controllers have been specifically designed for digital computer implementation, there exist a vast number of potential applications.

VI. REFERENCES

[1] P. R. KUMAR and A. BECKER, "A new family of optimal adaptive controllers", Math. Research Report No. 80-18, Sept. 1980, University of Maryland Baltimore County. Submitted for publication.

[2] P. MANDL, "Estimation and control in Markov Chains", Advances in Applied Probability, Vol. 6, pp. 40-60, 1974.

[3] V. BORKAR and P. VARAIYA, "Adaptive control of Markov Chains, I; finite parameter set", IEEE Transactions on Automatic Control, Vol. AC-24, pp. 953-958, 1979.

[4] P. R. KUMAR, "Adaptive control with a compact parameter set", Math. Research Report No. 80-16, University of Maryland Baltimore County, July 1980. Submitted for publication.

[5] B. DOSHI and S. SHREVE, "Strong consistency of a modified maximum likelihood estimator for controlled Markov Chains," Journal of Applied Probability, Vol. 17, 1980.

THE GLOBAL CONVERGENCE OF DIRECT ADAPTIVE INPUT MATCHING CONTROL OF SOME NONMINIMUM PHASE PLANTS

C. RICHARD JOHNSON, JR.*
Department of Electrical Engineering
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

G. C. GOODWIN** and K. S. SIN**
Department of Electrical and Computer Engineering
University of Newcastle
New South Wales, 2308, Australia

ABSTRACT

This paper establishes the global convergence (in the sense that the control input, plant output, and all adapting controller parameters remain bounded as the input and output asymptotically approach their respective optimal values) of adaptive implementation of one-step-ahead optimal controllers based on input matching and applied to all minimum phase and some nonminimum phase, linear, discrete-time, deterministic, time-invariant, single-input, single-output systems. The key assumption is that the one-step-ahead optimal controller designed using the true system parameters results in a stable closed-loop system.

I. INTRODUCTION

Only recently has the global convergence of adaptive controllers for single-input, single-output, discrete-time systems been proven [1]-[3]. This result establishes that the control input, plant output, and adapted controller parameters remain bounded as the performance of the controlled system asymptotically approaches that desired despite the initial bounded controller parameter estimate errors. One key assumption for these results is that the plant has a stable inverse, which is typically (mis)labeled as a minimum phase requirement.

This paper presents the first global convergence proof of a direct adaptive controller, which is applicable to some unstable, nonminimum phase plants. This proof follows that for the minimum phase case in [1]. The adaptive control scheme is based on the adaptive input matching implementation of one-step-ahead optimal control introduced in [4], which asymptotically minimizes

$$J[u(k-1)] = \tfrac{1}{2}\{[y^*(k) - y(k)]^2 + \lambda u^2(k-1)\} \tag{1}$$

by choice of $u(k-1)$. For the plant

---

* Supported by National Science Foundation Grant ECS-79-18577.

** Supported by Australian Research Grants Committee.

$$y(k) = \sum_{i=1}^{n} a_i \, y(k-i) + \sum_{j=1}^{m} b_j \, u(k-j) \tag{2}$$

(1) is minimized by the use of

$$u^*(k-1) = c_1 y^*(k) + \sum_{j=2}^{m} c_j \, u(k-j) + \sum_{i=1}^{n} d_i \, y(k-i), \tag{3}$$

where

$$c_1 = \frac{b_1}{b_1^{\,2}+\lambda}, \quad c_j = \frac{-b_j}{c_1}, \quad d_i = \frac{-a_i}{c_1} . \tag{4}$$

The effect of λ on the stability of the controlled system resulting from the use of (3) in (2) can be derived from a root locus interpretation [5] of the resulting characteristic equation

$$\left[\sum_{j=1}^{m} b_j \, z^{-j+1} \right] + \left(\frac{\lambda}{b_1} \right) \left[1 - \sum_{i=1}^{n} a_i \, z^{-i} \right] = 0. \tag{5}$$

For $\lambda = 0$, which results in the schemes of [1] and [6], (5) reduces to the numerator polynomial of (2), thereby requiring stable plant invertibility for a stable control system. As $\lambda \to \infty$, the roots of (5) become the poles of the plant in (1). Therefore the one-step-ahead optimal controller of (3) is broadly applicable given appropriate λ selection to plants which are stable or have a stable inverse. As shown in [4] and [5] this root locus interpretation reveals some unstable, nonminimum phase (i.e. unstably invertible) plants which are stabilized by (3) and appropriate λ and some which are not.

The next section presents the input matching solution to the adaptive control problem of directly estimating the control parameters in (3). The proposed input matching adaptive controller is proven in section 3 to be globally stable. The fourth section displays doubts concerning the extension of this method to the more general direct adaptive pole placement problem. The conclusion summarizes the advances afforded by this and related work.

II. DIRECT ADAPTIVE INPUT MATCHING SOLUTION

The direct adaptive control problem in asymptotically implementing the optimal control of (3) minimizing (1) can be viewed as the establishment of recursive formulas for updating estimates of the c_j and d_i in (3) used in forming

$$u(k-1) = \hat{c}_1(k-1) \, y^*(k) + \sum_{j=2}^{m} \hat{c}_j(k-1) \, u(k-j) + \sum_{i=1}^{n} \hat{d}_i(k-1) \, y(k-i) \tag{6}$$

such that

$$u(k-1) \to u^*(k-1) \quad \text{as} \quad k \to \infty \tag{7}$$

with $u(k)$, $y(k)$, $\hat{c}_j(k)$ for $j = 1, \ldots, m$, and $\hat{d}_i(k)$ for $i = 1, \ldots, n$ remaining bounded for all k. The solution suggested in [4] is based on the concept of input matching, i.e. after applying $u(k-1)$ observe $y(k)$, compare this with $y^*(k)$, and infer the optimal input $u^*(k-1)$ that should have been applied. Since

$$u^*(k-1) = \theta^T X(k-1), \tag{8}$$

where from (3)

$$\theta^T = [c_1 \; \ldots \; c_m \; d_1 \; \ldots \; d_n] \tag{9}$$

and

$$X^T(k-1) = [y^*(k) \;\; u(k-2) \; \ldots \; u(k-m) \;\; y(k-1) \; \ldots \; y(k-n)], \tag{10}$$

and

$$\hat{u}(k-1) = \hat{\theta}^T(k-1)\; X(k-1), \tag{11}$$

where from (6)

$$\hat{\theta}^T(k-1) = [\hat{c}_1(k-1) \; \ldots \; \hat{c}_m(k-1) \;\; \hat{d}_1(k-1) \; \ldots \; \hat{d}_n(k-1)], \tag{12}$$

given a posterior calculation of $u^*(k-1)$ the correction of $\hat{\theta}(k-1)$ is an equation error parameter estimation problem [7] as noted in [4]. Recognition of this structure suggests the following solution

$$\hat{\theta}(k) = \hat{\theta}(k-1) + [1 + X^T(k-1)\Gamma X(k-1)]^{-1}\, \Gamma X(k-1)[u^*(k-1) - u(k-1)] \tag{13}$$

where Γ is bounded, symmetric, and positive definite. Note from (8) and (11) that

$$u^*(k-1) - u(k-1) = \tilde{\theta}^T(k-1)\, X(k-1) = X^T(k-1)\tilde{\theta}(k-1) \tag{14}$$

where

$$\tilde{\theta}(\cdot) \triangleq \theta - \hat{\theta}(\cdot). \tag{15}$$

Implementation of (13) requires a causal formula for (14), which is suggested by the preceding input matching strategy and provided by

Lemma 1: The error between the optimal control input of (3) and the applied input of (6) is causally calculable given $y^*(k)$ and $y(k)$ prior to calculation of $\hat{\theta}(k)$ and $u(k)$ from

$$u^*(k-1) - u(k-1) = \left(\frac{b_1}{\lambda + b_1^2}\right)[y^*(k) - y(k) - (\frac{\lambda}{b_1})u(k-1)] \tag{16}$$

Proof: Refer to the proof of Lemma 2 in [4]. From (2)

$$y(k) - b_1 u(k-1) = \sum_{i-1}^{n} a_i y(k-i) + \sum_{j=2}^{m} b_j u(k-j). \tag{17}$$

Substitution of (17) into (3) given (4) yields

$$u^*(k-1) = (\lambda + b_1^2)^{-1}\, b_1[y^*(k) - y(k) + b_1 u(k-1)]. \tag{18}$$

subtracting $u(k-1)$ from both sides of (18) yields (16). □

Combination of (13) and (16) suggests the solution candidate

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \varepsilon[1+X^T(k-1)\,\Gamma X(k-1)]^{-1}\Gamma X(k-1)[y^*(k)-y(k)-\gamma u(k-1)] \tag{19}$$

where

$$\gamma = \frac{\lambda}{b_1}\,. \tag{20}$$

In [4] the satisfaction of (7) is based on the assertion that $X^T(k)\Gamma X(k)$, i.e. $X(k)$, remains bounded for all k. From (10) this is equivalent to the a priori assumption that $u(k)$ and $y(k)$ are bounded for all k, which is actually part of what is to be proven. The next section removes this undesirable assertion by use of the techniques in [1].

III. GLOBAL CONVERGENCE PROOF

The proof that under certain assumptions, to be summarized in the statement of the Main Theorem in this section, (7) results with $\hat{\theta}(k)$ and $X(k)$ bounded for all k from use of (19) in updating the controller parameters in (6) results from the use of the four following lemmas.

Lemma 2: If ε and γ are preselected such that

$$0 < \varepsilon(b_1 + \gamma) < 2 \tag{21}$$

then (19) results in

$$\lim_{k\to\infty} \frac{[u^*(k-1) - u(k-1)]^2}{1 + X^T(k-1)\Gamma X(k-1)} = 0. \tag{22}$$

Proof: Refer to Lemma 5.1 in [1] and Theorem A.1 in [4]. From (14)-(16) and (19)

$$\tilde{\theta}(k) = \tilde{\theta}(k-1) - \left[\frac{\varepsilon(\lambda+b_1^2)}{b_1}\right][1 + X^T(k-1)\Gamma X(k-1)]^{-1}\Gamma X(k-1)X^T(k-1)\tilde{\theta}(k-1). \tag{23}$$

Therefore

$$\begin{aligned}\tilde{\theta}^T(k)\Gamma^{-1}\tilde{\theta}(k) &- \tilde{\theta}^T(k-1)\Gamma^{-1}\tilde{\theta}(k-1)\\
&= \frac{\varepsilon(\lambda+b_1^2)}{b_1}[1+X^T(k-1)\Gamma X(k-1)]^{-1}\tilde{\theta}^T(k-1)X(k-1)\{-2\\
&\quad + \frac{\varepsilon(\lambda+b_1^2)}{b_1}[1+X^T(k-1)\Gamma X(k-1)]^{-1}X^T(k-1)\Gamma X(k-1)\}X^T(k-1)\tilde{\theta}(k-1)\\
&\le \{\frac{\varepsilon(\lambda+b_1^2)}{b_1}\}\{-2+\frac{\varepsilon(\lambda+b_1^2)}{b_1}\}[1+X^T(k-1)\Gamma X(k-1)]^{-1}\tilde{\theta}^T(k-1)X(k-1)X^T(k-1)\tilde{\theta}(k-1)\\
&\le 0
\end{aligned} \tag{24}$$

given (21). Since this negative definite difference must approach zero as $k \to \infty$ if $\tilde{\theta}(0)$ and Γ^{-1} are bounded and the first two terms in braces are bounded away from zero due to (21), then (22) results from (14). □

Lemma 3: The plant input u and output y are stably related to the input matching error $u^* - u$ via

$$A(q^{-1})[u^*(k-1)-u(k-1)] = \left(\frac{b_1}{\lambda+b_1{}^2}\right)\{A(q^{-1})y^*(k)-[\gamma A(q^{-1})+B(q^{-1})]u(k-1)\} \tag{25}$$

and

$$B(q^{-1})[u^*(k-1)-u(k-1)] = \left(\frac{b_1}{\lambda+b_1{}^2}\right)\{B(q^{-1})y^*(k)-[\gamma A(q^{-1})+B(q^{-1})]y(k)\} \tag{26}$$

where

$$A(q^{-1})x(k) = x(k) - \sum_{i=1}^{n} a_i x(k-i) \tag{27}$$

$$B(q^{-1})x(k) = \sum_{j=1}^{m} b_j x(k-j+1), \tag{28}$$

if γ is chosen such that

$$\gamma A(q^{-1}) + B(q^{-1}) = 0 \quad \text{iff } |q| < 1 \tag{29}$$

and $y^*(k)$ is bounded for all k.

Proof: From (16) in Lemma 1

$$\begin{aligned} &u^*(k-1) - u(k-1) - \sum_{i=1}^{n} a_i[u^*(k-i-1) - u(k-i-1)] \\ &= \left(\frac{b_1}{\lambda+b_1{}^2}\right)[y^*(k) - \sum_{i=1}^{n} a_i y^*(k-i) - \{y(k) - \sum_{i=1}^{n} a_i y(k-i)\} \\ &\qquad - (\frac{\lambda}{b_1})\, u(k-1) + (\frac{\lambda}{b_1}) \sum_{i=1}^{n} a_i u(k-i-1)]. \end{aligned} \tag{30}$$

But from (2)

$$y(k) - \sum_{i=1}^{n} a_i y(k-i) = \sum_{j=1}^{m} b_j u(k-j). \tag{31}$$

Therefore, using (20) and (31) in (30) yields (25). Similarly, from (16)

$$\begin{aligned} &\sum_{j=1}^{m} b_j[u^*(k-j) - u(k-j)] \\ &= \left(\frac{b_1}{\lambda+b_1{}^2}\right)[\sum_{j=1}^{m} b_j y^*(k-j+1) - \sum_{j=1}^{m} b_j y(k-j+1) \\ &\qquad - (\frac{\lambda}{b_1}) \sum_{j=1}^{m} b_j u(k-j)]. \end{aligned} \tag{32}$$

Substituting from (31) for the last term in (32) and using (20), (27), and

(28) yields (26). □

Lemma 4: Given the two-input, single-output system

$$F(q^{-1})y(k) = G(q^{-1})u_1(k) + H(q^{-1})u_2(k) \tag{33}$$

with $|u_1(k)| < \zeta^2 < \infty$ $\forall k$, $H(q^{-1}) \neq 0$ $\forall |q| \geq 1$, and F, G, and H are bounded operators then

$$|u_2(k)| \leq C_1 + C_2 \max_{0<\tau<k} |y(\tau)|. \tag{34}$$

Proof: Refer to Lemma 3.2 in [1]. Since u_1 and G are bounded the contribution of $G(q^{-1})u_1(k)$ to y is bounded and due to the stable invertibility of the u_2 to y transfer function (34) follows. □

Lemma 5: If

$$\lim_{k\to\infty} \frac{s^2(k)}{1 + \sigma(k)R\sigma(k)} = 0 , \tag{35}$$

where $\{s(k)\}$ is a real scalar sequence, $\{\sigma(k)\}$ is a real p-vector sequence, and R is a bounded, $p \times p$, real, symmetric, positive-definite matrix, and

$$\|\sigma(k)\| \leq C_3 + C_4 \max_{0<\tau<k} |s(k)| , \tag{36}$$

then

$$\lim_{k\to\infty} s(k) = 0 \tag{37}$$

and $\{\|\sigma(k)\|\}$ is bounded.

Proof: Refer to Lemma 3.1 in [1]. The proof remains unchanged with $b_1 = 1$ after adding the fact that

$$x^T\Gamma x \leq K\, x^T x \tag{38}$$

where K is the maximum eigenvalue of Γ from Theorem 8-17 of [8].

Successive use of the four preceding lemmas will result in the proof of the following

Main Theorem: Under the assumptions that (i) (2) exactly describes the plant, (ii) $y^*(k)$ in (1) is bounded for all k, (iii) u(k) is calculated and applied after exact $y^*(k)$ and y(k) measurement without affect on (2) or (3), (iv) γ is chosen in (19) such that (29) is satisfied, (v) ε is chosen, given appropriate γ, such that (21) is satisfied, and (vi) Γ in (19) is bounded, symmetric, and positive definite, the recursive adaption of the controller parameters in (6) via (19) will result in (7) and bounded u and y with (6) applied to (2) and thereby asymptotically lead to the minimization of (1) despite the bounded value of $\|\hat{\theta}(0)\|$.

Proof: From Lemma 2, (35) is satisfied with $s = u^* - u$, $\sigma = X$, and $R = \Gamma$. From Lemma 3 and 4 with assumption (iv), (36) is satisfied by simple summation of each term in (34) relating $u^* - u$ to the values of u and y in

(10) from (25) and (26) with the y^*, bounded from assumption (ii), in (10) absorbed into C_3. Therefore, given assumption (vi), (7) and bounded $\|X\|$ result. □

Remarks about assumptions:

(i) The principle requirement is that the autoregressive and moving-average lengths n and m and the system delay (or relative degree) of one (i.e. $b_1 \neq 0$) provide an exact plant description. If $b_1 = \ldots = b_{\ell-1} = 0$ and $b_\ell \neq 0$, then ℓ must be known. Typically n and m can be overly adequate but not insufficient for (7) and $y \to y^*$. There are presently no full stability results for reduced-order adaptive control, i.e. n and m in (6) are smaller than those of (2).

(ii) This standard assumption is required in several steps of the development and simply excludes the tracking of a ramp or other growing signals.

(iii) This rarely noted assumption is critical for sampled-data control where the plant description of (2) is affected by an appreciable lag between y(k) measurement and u(k) application. The additional open problems of sampled-data adaptive control have rarely been considered. A recent paper [9] examines a continuous-time plant-controller combination with discrete-time adjustment of the controller parameters. Also, obvious from (6) is the need for y^* "prediction" in the sense that $y^*(k)$ is required in u(k-1). This can be avoided by altering (1) to include $[y^*(k-1) - y(k)]^2$ (or $[y^*(k-\ell)-y(k)]^2$) rather than $[y^*(k) - y(k)]^2$.

(iv) The proper a priori choice of γ simply requires the reasonable in that the control objective, e.g. minimization of (1), results in a stable control system if the a_i and b_j of (2) were used in (4) thereby resulting in (5). This requirement is also the main drawback to this implementation of input matching, since no compact strategy exists for γ selection given limited a priori knowledge of the plant coefficients.

(v) For $\lambda = 0 \Rightarrow \gamma = 0$, this requirement reduces to the standard assumption that the sign of b_1 and some upper bound on its magnitude are known. For $\lambda \neq 0$, if $\operatorname{sgn}(b_1) \neq \operatorname{sgn}(\gamma)$ then (21) displays a looser bound on b_1 foreknowledge. However, the interconnection with the possibly restricted γ is obvious.

(vi) Commonly Γ is diagonal, which uncouples the parameter recursions in (19) and reduces the computational load. Particular types of time-varying, e.g. decaying, Γ can also be incorporated [7].

IV. EXTENSION TO POLE PLACEMENT

Modifying an incremented version of the control law of (3) to compose the present u of past u and y (or effectively state values) as

$$u(k) = \delta y^*(k+1) + \sum_{i=1}^{n} [c_i u(k-i) + d_i y(k-i)] \tag{39}$$

allows arbitrary pole placement as recognized in [10]. Considering y^* as a reference input and substituting (3) into (2) yields

$$y(k) = \left\{ \frac{\delta q^{-1} B(q^{-1})}{C(q^{-1})A(q^{-1}) - q^{-1}B(q^{-1})D(q^{-1})} \right\} y^*(k+1) \tag{40}$$

where

$$C(q^{-1})x(k) = x(k) - \sum_{i=1}^{n} c_i x(k-i) \tag{41}$$

$$D(q^{-1})x(k) = \sum_{i=1}^{n} d_i x(k-i). \tag{42}$$

For the preceding special case, appropriate selection of the δ, C, and D can yield the characteristic equation $\gamma A(q^{-1}) + B(q^{-1})$ of (5), (25) and (26). Attempting to logically extend the input matching error of (16) to

$$\begin{aligned} u^*(k-1) - u(k-1) &= \alpha y^*(k) - \sum_{i=0}^{n} p_i y(k-i) - \sum_{j=1}^{n} q_j u(k-j) \\ &= \alpha y^*(k) - P(q^{-1})y(k) - Q(q^{-1})u(k) \end{aligned} \tag{43}$$

appears to afford the possibility of extending the input matching concept to the open problem of direct adaptive pole placement, if P and Q can be chosen without exact a priori knowledge of A and B as γ can be in (19).

Attempting to repeat (25) - (32) leads to

$$\begin{aligned} &A(q^{-1})\ [u^*(k-1) - u(k-1)] \\ &\quad = \alpha A(q^{-1})y^*(k) - [q^{-1}P(q^{-1})B(q^{-1}) + Q(q^{-1})A(q^{-1})]u(k) \end{aligned} \tag{44}$$

and

$$\begin{aligned} &q^{-1}B(q^{-1})[u^*(k-1) - u(k-1)] \\ &\quad = \alpha q^{-1}B(q^{-1})y^*(k) - [q^{-1}P(q^{-1})B(q^{-1}) - Q(q^{-1})A(q^{-1})]y(k), \end{aligned} \tag{45}$$

since from (31)

$$A(q^{-1})y(k) = q^{-1}B(q^{-1})u(k). \tag{46}$$

From (44) and (45) P and Q must be chosen such that

$$\Lambda(q^{-1}) = q^{-1}P(q^{-1})B(q^{-1}) - Q(q^{-1})A(q^{-1}) = 0 \quad \text{iff } |q| < 1 \tag{47}$$

in order to follow the same development as in the preceding section. By comparison with (40), this requires that

$$P(q^{-1}) = D(q^{-1}) \tag{48}$$

$$Q(q^{-1}) = C(q^{-1}), \tag{49}$$

i.e. the answer must be known before (43) can be formed. This absurdity results since the polynomial Λ relating $u^* - u$ to u and y is the characteristic equation of the control system resulting from exact a priori knowledge of A and B. Compare (5) with (29) for the one-step-ahead optimal case. Reflection on the underlying block diagram structure will substantiate this equivalence. Therefore, if P and Q could be chosen for (43) the desired

C and D would be known, which would obviate adaption. This appears to preclude simplistic extension of the input matching concept for direct adaptive pole placement.

V. CONCLUSION

The principal contributions of [4] are the recognition of Lemmas 1 and 2 and the proposal of (19), which can be derived by alternate interpretation, e.g. [11]. The major contribution of [1] to the present paper is embodied in Lemmas 4 and 5. This paper adds Lemma 3 to yield the Main Theorem, which provides the first global stability proof of a direct adaptive controller applicable to some nonminimum phase systems, i.e. that set for which γ can be chosen a priori such that (29) is satisfied.

The deterministic results in this paper have been extended in [12] to the case where (1) includes a filtered version of u, e.g. incorporating integral action as suggested in [11] to remove DC tracking offset, and to the stochastic case where colored noise appears in the plant equation. Despite the gloomy prediction in section IV regarding extension of this technique to direct adaptive pole placement, this paper provides the first proof of global stability, without sufficient excitation assumptions as in [13], of an adaptive controller applicable to some linear, time-invariant, nonminimum phase plants with unknown coefficients.

VI. REFERENCES

[1] G. C. Goodwin, P. J. Ramadge, and P. E. Caines, "Discrete-time multivariable adaptive control," IEEE Trans. on Auto. Control, vol. AC-25, no. 3, pp. 449-456, June 1980.

[2] K. S. Narendra and Y.-H. Lin, "Stable discrete adaptive control," IEEE Trans. on Auto. Control, vol. AC-25, no. 3, pp. 456-461, June 1980.

[3] B. Egardt, "Stability analysis of discrete-time adaptive control schemes," IEEE Trans. on Auto. Control, vol. AC-25, no. 4, pp. 710-716, August 1980.

[4] C. R. Johnson, Jr. and E. Tse, "Adaptive implementation of one-step-ahead optimal control via input matching," IEEE Trans. on Auto. Control, vol. AC-23, no. 5, pp. 865-872, October 1978.

[5] C. R. Johnson, Jr., "On single-stage optimal control," Proc. 1978 IEEE Southeastcon, Atlanta, GA, pp. 511-514, April 1978.

[6] K. J. Åström and B. Wittenmark, "On self-tuning regulators," Automatica, vol. 9, no. 2, pp. 185-199, March 1973.

[7] C. R. Johnson, Jr., "Adaptive parameter matrix and output vector estimation via an equation error formulation," IEEE Trans. on Sys., Man, and Cybernetics, vol. SMC-9, no. 7, pp. 392-397, July 1979.

[8] C. T. Chen, Introduction to Linear System Theory. New York: Holt, Rinehart and Winston, p. 340, 1970.

[9] H. Elliott, "Hybrid adaptive control of continuous time systems,"

Colorado St. Univ. Dept. of Elec. Eng. Tech. Report No. JL80-DELENG-1, July 1980.

[10] C. R. Johnson, Jr., "On direct adaptive control of nonminimum-phase plants," Info. Sciences, vol. 20, pp. 165-179, April 1980.

[11] D. W. Clarke and P. J. Gawthrop, "Self-tuning control," Proc. IEE, vol. 126, no. 6, pp. 633-640, June 1979.

[12] G. C. Goodwin, C. R. Johnson, Jr., and K. S. Sin, "Global convergence for adaptive one-step-ahead optimal control based on input matching," Univ. of Newcastle Dept. of Elec. Eng. Tech. Report No. EE 8006, February 1980.

[13] G. Kreisselmeier, "Adaptive control via adaptive observation and asymptotic feedback matrix synthesis," IEEE Trans. on Auto. Control, vol. AC-25, no. 4, pp. 717-722, August 1980.

THE USE OF ADAPTIVE CONTROL FOR HELICOPTER TRAJECTORIES IN SEARCH OPERATIONS

A. Chacón*, R. Padilla**, C. Padilla**, M. Azagoury*, B. Fernández*

* Universidad Simón Bolívar
Departamento de Procesos y Sistemas,
Caracas, Venezuela

** Instituto Venezolano de Investigaciones Científicas (IVIC)
Laboratorio de Ingeniería de Sistemas,
Apartado 1827, Caracas 1010 A, Venezuela

ABSTRACT

An effective and successful search operation by helicopter requires close tracking of predetermined trajectories defined by the characteristics of the terrein and the prior information available to the search party. We investigate here the design of an adaptive controller such that the helicopter follows a predetermined trajectory in a horizontal plane. A model reference control approach is used. Simulation results are given to illustrate the behavior of the controller.

I. Introduction

The tracking of the predetermined search trajectory in helicopter rescue operations is an important ingredient in a successful operation. The trajectories are designed based on the characteristics of the terrein and the prior information available about the unknown location of the objetive. The trajectories are designed so that an efficient and thorough covering of a given area is achieved.

Usually the pilot is charged with the responsability of maintaining the actual trajectory as close as possible to the designed search trajectory. An automated check on how closely the designed search trajectory is followed and an automatic correction of the deviations is an important aid to the success of the search. The control of the helicopter is achieved thru the direct manipulation of five control variables: throttle, collective pitch, longitudinal cyclic, lateral cyclic, and tail rotor pitch. In actual operation the throttle is automatically actuated to maintain a constant main rotor speed, and the tail rotor pitch is directly actuated by the pilot to maintain the yaw rate within prescribed limits. The remaining input variables used for search trajectory tracking are the collective pitch and the two cyclics. Two typical search trajectories are utilized. The square wave type trajectory , most useful for plain terrein and when there is little information about the possible location of the objetive, and the square type used for mountainous areas or when there are indications of the possible location of the objetive.

The helicopter model used both in the design of the adaptive control and the simulation is represented by a set of nonlinear equations /1/, /2/.

$$\dot{x} = f\ (x,\ u)$$

where, x is the state vector formed by the three body axis velocities, three body axis rotational rates and the three Euler angles of the vehicle relative to a ground frame of reference, and u is the vector of four control inputs collective pitch, the two cyclics, and tail rotor pitch. In the simulation, technical data corresponding to Bell QD1 helicopter was used.

As the helicopter tracks a given search trajectories it goes through a series of changes in its modes of operation that can be taken into account to advantage in performance by an adaptive controller. The relative simplification that is obtained for trajectories in a horizontal plane, like the ones of interest in this work, it is expected to arrive at implementable designs for on line adaptive controllers.

An explicit model reference adaptive controller /3/ is used where the model to be tracked is determined by the simplified linear dynamics obtained for the particular trajectories of interest. The performance of the controller is evaluated by simulation using the full nonlinear dynamics as a representation of the actual vahicle.

II. Problem Description

This section is devoted to describing the helicopter equations of motion and the particular trajectories of interest in this work.

II.1 Helicopter Equations of Motion.

The equations of motion of the helicopter are obtained through direct application of Poisson's formulas and Euler equations. See appendix A for a list of the symbols used for the various variables entering in the description of the helicopter dynamics. For details on the various sets of coordinate axes used see Ref. /1/ and /2/.

With these notations and references the equations of motion of the aircraft are:

$$\dot{p} - (L_A-(I_{yy}-I_{zz})\ qr)/I_{xx} \quad , \tag{1}$$

$$\dot{q} = (M_A-(I_{zz}-I_{xx})\ rp)/I_{yy} \quad , \tag{2}$$

$$\dot{r} = (N_A-(I_{xx}-I_{yy})\ pq)/I_{zz} \quad , \tag{3}$$

$$\dot{\phi} = p - \psi \sin \Theta \ , \tag{4}$$

$$\dot{\Theta} = r \sin \phi - q \cos \phi \ , \tag{5}$$

$$\dot{\psi} = (r \cos \phi + q \sin \phi)/\cos \Theta \quad , \tag{6}$$

$$\dot{u} = x_A/M + r\Theta - qw - g \sin \Theta \quad , \tag{7}$$

$$\dot{v} = y_A/M + pw - ru + g \cos \Theta \ \sin \phi \quad , \tag{8}$$

$$\dot{w} = z_A/M + qu - p\Theta \ + g \cos\Theta \cos \phi \ . \tag{9}$$

The Forces and Torques acting on the aircraft are given by :

$$x_A = F_{RP} \cos \beta \sin \alpha \tag{10}$$

$$y_A = F_{RP} \sin\beta - F_{RC} \tag{11}$$

$$z_A = -F_{RP} \cos\beta \tag{12}$$

$$L_A = F_{RP} \sin\beta \;\; z_{RP} - F_{RC}\, z_{RC} + M_{RP} \cos\beta \;\; \sin\alpha \quad , \tag{13}$$

$$M_A = M_{RP} \sin\beta + M_{RC} + F_{RP} \cos\beta \;\; \cos\alpha \;\; x_{RP} \tag{14}$$

$$+ F_{RP} \cos\beta \;\; \sin\alpha \;\; z_{RP} \quad ,$$

$$N_A = F_{RC}\, x_{RC} + F_{RP} \sin\beta \;\; x_{RP} - M_{RP} \cos\beta \;\; \cos\alpha \quad . \tag{15}$$

The determination of the effect of the pilot's actions on the Forces and Torques is next. The main variable in this problem is the angle of attack of the blades of the rotor, be it of the main rotor α_{RP} of the tail rotor, α_{RC}. These are determined by the mechanical linkages from the pilots actions through

$$\alpha_{RP} = \lambda_0 + K_L \lambda \;, \tag{16}$$

$$\alpha_{RC} = \delta_0 + D_D \delta \;, \tag{17}$$

where $\lambda_0 = 37.53$, $K_L = 0.52$, $\delta_0 = 17.18$, and $K_D = 1.05$, and λ and δ are the pilot's commands. From these, the magnitud and directions of the forces and torques may be immediately determined by aerodynamic considerations as

$$F_{RP} = \frac{2}{3} \pi (RPM_{RP})^2 R_{RP}^3 \;\; L_{RP}\, \alpha_{RP} \sin \;\alpha_{RP} \quad , \tag{18}$$

$$F_{RC} = \frac{2}{3} \pi (RPM_{RC})^2 R_{RC}^3 \cdot L_{RC}\, \alpha_{RP} \sin \alpha_{RC} \quad , \tag{19}$$

$$M_{RP} = \frac{\pi}{2} (RPM_{RP})^2 \cdot R_{RP}^4 \cdot L_{RP} \cdot \alpha_{RP} \frac{\sin^2 \alpha_{RP}}{\cos \alpha_{RP}} \;, \tag{20}$$

$$M_{RC} = \frac{\pi}{2} (RPM_{RC})^2 R_{RC}^4 \cdot L_{RC} \cdot \alpha_{RC} \frac{\sin^2 \alpha_{RC}}{\cos \alpha_{RC}} \quad , \tag{21}$$

where R_{RP}, R_{RC}, L_{RP}, L_{RC} are the aerodynamical parameters of the blades of the main and tail rotors. The directions are defined by

$$\alpha = \text{arc tg } (\cos \Theta_H) \;, \tag{22}$$

$$\beta = \text{arc} \quad \cos \left[\sin \Theta_{IN} \sqrt{1+\cos^2 \Theta_H}\right] \tag{23}$$

This completes the description of the dynamical equations of the helicopter.

II.2 Search Trajectories.

The trajectory considered in this paper is the square wave type. This trajectory is defined on a horizontal plane, and it has two sections; A linear part where the flight conditions are:

$u = 50$ knots

$\phi = 0$

$\Theta = 0.174$ rad

$p = 0 \quad , \quad q = 0$

and turn of 90° where the helicopter behave as;

$u = 50$ knots

$\phi = 0.174$ rad

$\Theta = 0.174$ rad

$p = 0.5$ rad/sec

$q = 0$

III. Design and Implementation of the Adaptive Controller

III.1 The MRA controller.

The model reference adaptive techniques (MRAS) with an explicit model was used to design the adaptive controller. The block diagram of the control system is shown in figure 1

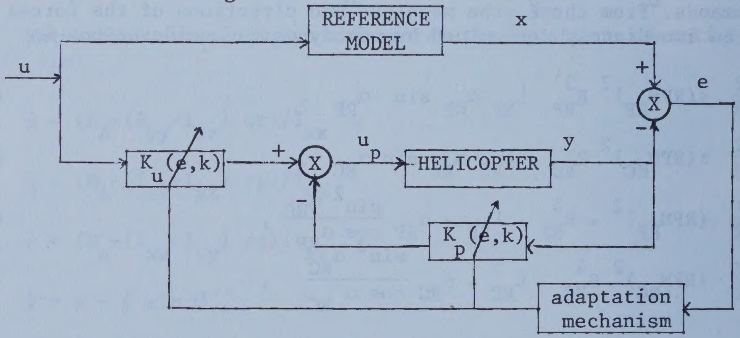


FIGURE 1

where $K_u(e,k)$ and $K_p(e,k)$ are the adaptive feedforward and feedback gain matrices, which are on line by the modified adaptation mechanism as a function of the generalized state error vector $\underline{e}$, given by

$$\underline{e} = \underline{x} - \underline{y} \quad . \tag{24}$$

The design objective will be to find an adaptive law such that the gain matrices $K_u(e,k)$ and $K_p(e, k)$ are modified so that $\underline{e}$ tends toward $\underline{o}$ for any input $\underline{u}$ guaranteeing certain index of performance which in this case it is specified by the dynamic behavior of the reference model.

III.2 The Reference Model.

As we established above, a reference model specifying the desired dynamic behavior of the helicopter must be found in order to design the adaptive controller. This model must satisfy the following two basic conditions:

1. It must be a time-invariant linear system.
2. Its dimension must be the same as the system to be controlled.

In reference /2/ a linearized set of state space equations describing the helicopter motion were obtained applying perturbation techniques to the nonlinear set of equations given in section II. As we can see from this paper, the elements of the state matrix A are variable depending on the flight conditions and the desired trajectory.

In our case, we choose the reference model to have the same structure as the linearized state space system obtain in reference /1/, but in order to satisfy condition (1) two models where considered according to the two types of flight trajectory under study, say stationary flight and turn of 90° degrees. We assume in our design that only the angle ϕ changes from one trajectory to another. The values of the element of the matrices A_M and B_M of the reference models where found by identification under two types of situations:

i) From hoover operation to level flight.
ii) From level flight to turn of 90° degrees.

The values of the matrices A_M and B_M using only a reduced 4 order model are shown in Table II.1

Level Flight

$$\begin{bmatrix} 0.58194 & 0 & 0 & 0 \\ -0.04196 & 0.82502 & 0 & 0 \\ 0.03861 & 0 & 0.9 & 0 \\ 0 & 0.044576 & 0 & 0.9 \end{bmatrix}$$

Matrix A_M

$$\begin{bmatrix} -0.1627 & 0.0539 & 0.084 \\ 0.075 & -0.06 & -0.0027 \\ -0.11627 & -0.0539 & -0.087 \\ 0 & 0 & 0 \end{bmatrix}$$

Matrix B_M

For the turn of 90° the matrices A_M and B_M are given by :

Turn of 90°

$$\begin{bmatrix} 0.58194 & 0 & 0 & 0 \\ -0.04196 & 0.82502 & 0 & 0 \\ 0.0386 & -0.0013919 & 0.9 & 0 \\ -0.0011632 & 0.044576 & 0 & 0.9 \end{bmatrix}$$

Matrix A_M

$$\begin{bmatrix} 0.11627 & 0.0539 & 0.084 \\ 0.075 & -0.06 & -0.0027 \\ -0.11397 & -0.05581 & -0.08413 \\ -0.05352 & 0.058852 & 0.0027 \end{bmatrix}$$

Matrix B_M

Some elements of the matrix A_M where changed, experimentaly, from the ones obtained by the identification, to improve the transient response and to satisfy the perfect model following condition required by the adaptation mechanism.

III.3 Adaptation Mechanism.

The adaptation mechanism, studied in reference /3/, is obtained applying the hyperstability theorem to find a law such that the state error $\underline{e}$ tends toward $\underline{o}$ according to a pre-specified dynamic behavior given by the reference model.

The algorithm is summarized as follows:

Plant input:

$$u_p(k) = -K_p(e, k)y_k + K_M x_k + K_u(e, k)\, u_M(k) \tag{25}$$

$$K_p(e, k) = K_p - \Delta K_p(e, k) \tag{26}$$

$$K_u(e, k) = K_u + \Delta K_u(e, k) \tag{27}$$

or

$$u_p(k) = u_{p1}(k) + u_{p2}(k) \tag{28}$$

$$u_{p1}(k) = -K_p y_k + K_u u_M(k) + K_M x_k \tag{29}$$

$$u_{p2}(k) = \Delta K_p^I(k) y_k + \Delta K_u^I(k)\, u_M(k) \tag{30}$$

Reference Model

$$\tilde{x}^0_{k+1} = A_M \tilde{x}_k + B_M u_M(k) \tag{31}$$

$$\tilde{x}_{k+1} = A_M \tilde{x}_k + B_M u_M(k) - B_p \left[\Delta K_p(k+1) - \Delta K_p^I(k)\right] y_k \tag{32}$$

$$-B_p \left[\Delta K_u(k+1) - \Delta K_u^I(k)\right] u_M(k)$$

where K^I means integral gain , x^0 means a priori value.

Plant to be controlled (helicopter)

$$y_{k+1} = A_p y_k + B_p u_p(k) \tag{33}$$

Generalized state error

$$e_k^0 = \tilde{x}_k^0 - y_k \tag{34}$$

$$e_k = \tilde{x}_k - y_k \tag{35}$$

Adaptation algorithm

$$V_{k+1}^0 = D\, e_{k+1}^0 \tag{36}$$

$$\begin{aligned} V_{k+1} &= \{I + DB_p\left[(F+F')\, y_k^T G y_k + (M+M')\, u_M^T(k)\, N u_M(k)\right]\}^{-1}. \\ &\quad V_{k+1}^0 \end{aligned} \tag{37}$$

$$\Delta K_p(k+1) = \Delta K_p^I(k+1) + \Delta K_p^P(k+1) \tag{38}$$

$$\Delta K_p^I(k+1) = \Delta K_p^I(k) + FV_{k+1}\, \left[G y_k\right]^T \tag{39}$$

$$\Delta K_p^P(k+1) = F' V_{k+1}\, \left[G y_k\right]^T \tag{40}$$

$$F_1 G > 0 \;;\; F' \geq 0$$

$$\Delta K_u(k+1) = \Delta K_u^I(k+1) + \Delta K_u^P(k+1) \tag{41}$$

$$\Delta K_u^I(k+1) = \Delta K_u^I(k) + MV_{k+1}\, \left[N u_M(k)\right]^T \tag{42}$$

$$\Delta K_u^P(k+1) = M' V_{k+1}\, \left[N u_M(k)\right]^T \tag{43}$$

$$M, N > 0 \;;\; M' \geq 0$$

Perfect model following conditions

$$(I - B_p B_p^+)\,(A_M - A_p) = 0 \qquad K_u = B_p^+ B_M$$

$$(I - B_p B_p^+)\, B_M = 0 \qquad K_p = K_M - B_p^+\,(A_M - A_p)$$

B_p^+ is the Penrose pseudo - inverse of B_p

Stability conditions

$$D\, z\, (zI - A_M + B_p K_M)^{-1}\, B_p \quad \text{strictly positive real transfer matrix, or} \tag{45.a}$$

$$D = B_p^T P\,,\ P > 0 \quad \text{where } P \text{ satisfies} \tag{45.b}$$

$$(A_M - B_p K_M)^T\, P\, (A_M - B_p K_M) - P = -Q, \quad Q > 0 \tag{46}$$

The matrices F, M and F', M' are the integral and proportional gain matrices which are choosen in such a way of increasing the speed of adaptation of the algorithm and at the same time to avoid large transients during the transient state.

The matrix D, named the linear compensator matrix, is found to satisfy the stability condition given in (45-46). The linear and constant feedforward and feedback matrices K_u, K_p are adapted for the system to follow the reference model.

In our design we have used two differents models depending on what type of trajectory we want to control the helicopter, which implies that two different set of values of the matrices D, K_p, K_u must be used in the adaptation mechanism. The change from one set of values to another is made on line according to the type of trajectory. In the next section, we summarize the values of the different matrices used by the adaptation mechanism.

IV. Simulation Results

An aproximate solution, given in /3/, of the algorithm of section III.3 was used to reduce computational complexity. For this case state equation for the model and the input $u_{p2}(k)$ are given as follows:

$$\underline{x}_{k+1} = A_M x_k + B_M u_M(k) \tag{47}$$

$$u_{p2}(k) = \Delta K_p(k+1) y_k + \Delta K_u(k+1) u_M(k) \tag{48}$$

where $\Delta K_p(k+1)$ and $\Delta K_u(k+1)$ are predicted adaptive gains which are obtained from eq. (37) substituting V^0_{k+1} by V_k. The gain matrices F, M, F', M' and Q were chosen to be diagonal and given by:

F = diag (5, 5, 1) , M = diag (5, 5, 1)
F'= diag (1, 1, 5) , M'= diag (1, 1, 5)
Q = diag (5, 5, 1, 1) .

The matrices K_u and K_p are

Level flight

$$K_u = I$$

$$K_p = \begin{bmatrix} -0.0001977 & 0.011026 & 0 & 9.6 \\ -0.001984 & 0.11032 & 0 & 9.6 \\ -0.10653 & -0.05518 & -0.0001024 & -51.2 \end{bmatrix}$$

Turn flight

$$K_u = I$$

$$K_p = \begin{bmatrix} 0.000338 & 0.00225 & 0 & 0 \\ -0.0046433 & -0.0023589 & -102{,}4 & 0 \\ -0.0003328 & -0{,}0022202 & 0 & 0 \end{bmatrix}$$

The control inputs to the helicopter were constrained according to the table IV.1 /2/ :

Control Constraints

Table IV.1

control	upper limit (radians)	lower limit (radians)
u_1	0.122	- 0.189
u_2	0.23	- 0.230
u_3	0.401	- 0.122

where the nominal values of the inputs for the model are for level flight: $u_1 = 0.0$, $u_2 = -0.2$, $u_3 = 0.0$ and for turn flight: $u_1 = -0.1$, $u_2 = 0.2$, and $u_3 = 0.0$.

Figures 2, 3, 4, 5, show the state generalized errors and the helicopter inputs for the two types of trajectories considered. We can see from these figures that the state errors pitch and roll never go to zero but they stay around a value of 0.2 rad/sec. after 5 seconds of running time.

V. Concluding Remarks

In this paper we investigate the design of an adaptive model reference control for a helicopter searching trajectories. The nonlinear system and the adaptive controller was simulated in a microcomputer. For a practical implementation of this control in a real helicopter, additional work for reducing the computational complexity has to be made since the original algorithm requires a lot of time to find the adaptive gains on line.

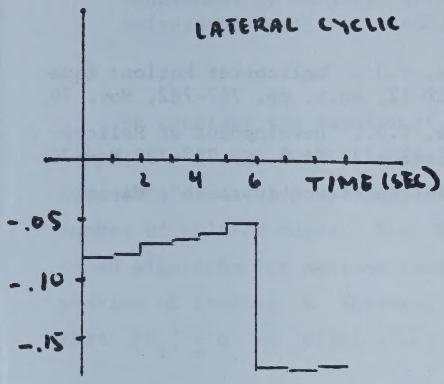


FIG. 2

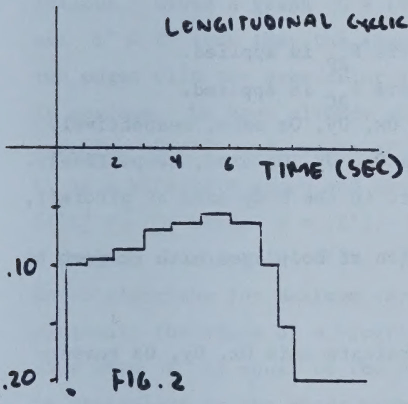


FIG. 2

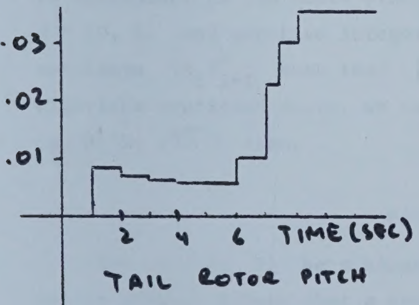


FIG. 3

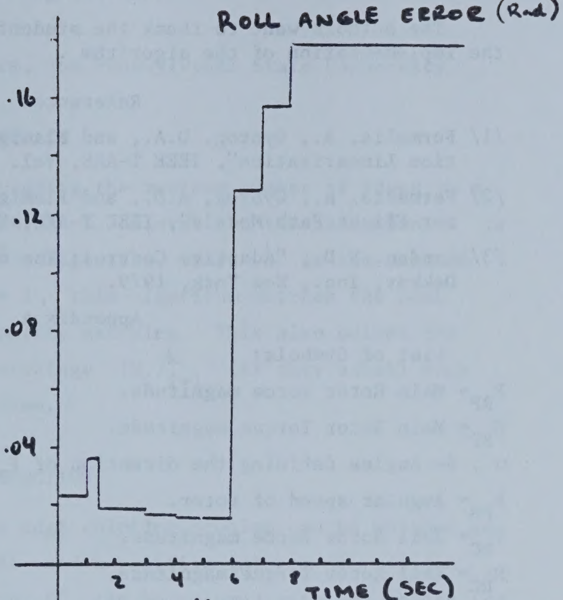


FIG. 4

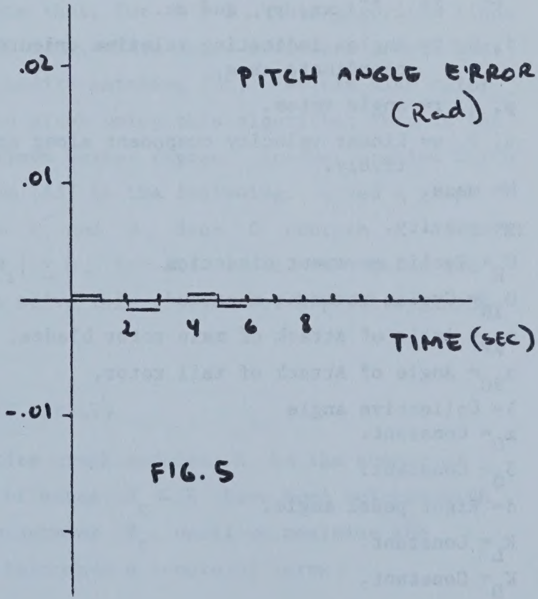


FIG. 5

Acknowledgments

The authors want to thank the student Gerardo Fernández for his help in the implementation of the algorithm .

References

/1/ Fermelia, A., Gyorog, D.A., and Flamigan, V.J., "Helicopter Motion: Equation Linearization", IEEE T-AES, Vol. AES-12, No.6, pp. 767-782, Nov. 76.

/2/ Fermelia, A., Gyorog, A.D., and Flamigan, V.J., "Development of Helicopter Flight Path Models", IEEE T-AES, Vol.AES-12, No.6, pp.783-792,Nov.76.

/3/ Landau, Y.D., "Adaptive Control: The model reference approach", Marcel Dekker, Inc., New York, 1979.

Appendix A

List of Symbols:

F_{RP}= Main Rotor Force magnitude.

M_{RP}= Main Rotor Torque magnitude.

α , β= Angles defining the direction of F_{RP} and M_{RP}

R_{PM}= Angular speed of rotor.

F_{RC}= Tail Rotor Force magnitude.

M_{RC}= Tail Rotor Torque magnitude.

x_{RP} , z_{RP}= x & z coordinates of the point where F_{RP} is applied.

x_{RC} , z_{RC}= x & z coordinates of the point where F_{RC} is applied.

x_A, y_A, z_A= resultant force components along 0x, 0y, 0z axes, respectively.

L_A, M_A, N_A= resultant torque components along 0x, 0y, 0z axes, respectively.

I_{xx}, I_{yy}, I_{zz} = Moments of Inertia with respect to the body axes of aircraft, xx, yy, and zz.

ϕ, Θ, ψ= Angles indicating relative orientation of body axes with respect to coordinate axes.

p, q, r= Angle rates.

u, v, w= Linear velocity component along coordinate axis 0x, 0y, 0z respectively.

M= mass.

g= gravity.

Θ_H= Cyclic movement direction.

Θ_{IN}= Cyclic movement magnitud.

α_{RP}= Angle of Attack of main rotor blades.

α_{RC}= Angle of Attack of tail rotor.

λ= Collective angle

x_0= Constant.

δ_0= Constant.

δ= Right pedal angle.

K_L= Constant.

K_D= Constant.

K-COLORING OF BIPARTITE GRAPHS

V. ASHOK AND JOSEPH JA'JA'
Department of Computer Science, The Pennsylvania State University
University Park, PA 16802

ABSTRACT

We consider the problem of coloring the maximum number of edges in a graph $G = (V, E)$ with a given set of K colors. In the case when G is bipartite we develop an $O(|E| \sqrt{s})$ algorithm, where s is the maximum number of colored edges. For $K = 1$, this algorithm matches the best known algorithm for maximum cardinality matching. This also solves the problem of finding k disjoint matchings $\{M_i\}_{i=1}^{k}$ (if they exist) such that $|M_i| \geq \alpha$ in $O(|E| \sqrt{k\alpha})$ time.

INTRODUCTION

An interesting variant of the edge coloring problem can be phrased as follows. Given a graph $G = (V, E)$ and a positive integer k, find a set $E' \subseteq E$ such that the edges in E' can be colored with k colors (no two edges with the same color should share a vertex) and such that $|E'|$ is maximum. We have already shown [AJ] that this problem is NP-complete even for $k = 2$ and graphs of maximum degree 3. We consider the case when G is a bipartite graph and develop an algorithm with running time $O(|E| \sqrt{s})$, where $s = |E'|$. Note that, for $k = 1$, this algorithm finds a maximum matching in $O(|E| \sqrt{n})$ time, which is as efficient as the best known algorithm for maximum cardinality matching [HK]. We can also color optimally the edges of a bipartite graph using this algorithm; this is the case when k is equal to the maximum vertex degree. Another problem which is equivalent to the above problem [AJ] is the following. Given a graph $G = (V, E)$ and positive integers k and α, does G contain k disjoint matchings $\{M_i\}_{i=1}^{k}$ such that $|M_i| \geq \alpha$, for all $1 \leq i \leq k$? Using the algorithm mentioned above, we can solve this problem for bipartite graphs in $O(|E| \sqrt{K\alpha})$ time.

BASIC RESULTS

Let $G = (V, E)$ be a bipartite graph and let k be the number of colors given. Assume that a set of edges $E_c \subseteq E$ have been colored with these k colors. We will try to augment E_c until we maximize the number of colored edges. We now introduce a couple of terms.

Let $E_b = E - E_c$. We will refer to E_c and E_b as the set of colored and blank edges (respectively) at a given stage. Let $V_0 = \{u \epsilon V | k$ edges in E_c are incident on $u\}$ and let $V_c = V - V_0$. We say that a vertex $u \epsilon V_c$ has <u>color c_1 missing</u> if no edges of color c_1 is incident on v. A path $P = \{e_1, e_2, \ldots, e_i\}$ of G is called a <u>color alternating path</u> if the edges of P alternate from E_b and E_c. A path $P = \{e_1, e_2, \ldots, e_{2i+1}\}$ is a <u>color augmenting path</u> (or simply an augmenting path) if

(a) P is a color alternating path

(b) $e_1, e_{2i+1} \; \epsilon \; E_b$

(c) the endpoints of P are in V_c.

Given an augmenting path $P = \{e_1, e_2, \ldots, e_{2i+1}\}$, the following procedure increases the number of colored edges.

```
procedure AUGMENT(G, P)
//G is a bipartite graph and P is a color augmenting path with
respect to a k-coloring C//
    Let $P = \{e_1, e_2, \ldots, e_{2i+1}\}$
    for j ← 1 to 2i - 1 by 2 do
        ROTATE($e_j$, $e_{j+1}$)
    endfor
    call AUG($e_{2i+1}$)
end AUGMENT
```

The procedures ROTATE and AUG are given below.

```
procedure ROTATE((u, v), (v, w))
//Note that $(u, v) \epsilon E_b$, $(v, w) \epsilon E_c$ and $u \epsilon V_c$//
    $c_1$ ← color missing at u ;
    $c_2$ ← color of edge (v, w) ;
    If $c_1 = c_2$ then [color (u, v) with $c_1$ and make $(v, w) \epsilon V_b$]
        else [Let $P_v$ and $P_w$ be maximal paths (or a circuit) with
              edges color alternately with $c_1$ and $c_2$ starting
              respectively at v and w ;
              //note that either $P_v$ or $P_w$ or both could be empty//
              Interchange colors on $P_v$ and $P_w$ (or circuit);
              Color (u, v) with $c_1$ and make $(v, w) \epsilon V_b$ ;
              //w has now the color $c_1$ missing//]
end ROTATE
```

AUG is just the procedure of Gabow and Kariv [GK] that colors an edge in E_b with colors missing at both ends.

```
procedure AUG((u, v))
    $c_1 \leftarrow$ color missing at u ;
    $c_2 \rightarrow$ color missing at v ;
    Let P' be a maximal path with edges alternately colored $c_1$ and
    $c_2$ starting at u (or v).
    Interchange colors in P' ;
    //(u, v) can now be colored with $c_1$ or $c_2$//
    Color (u, v) with $c_1(c_2)$.
end AUG
```

Actually, one can prove that the following is true [AJ].

Theorem 2.1: Let E be a set of colored edges with k colors in a bipartite graph G. Then $|E_c|$ is maximum iff G contains no color augmenting paths.

Therefore, an obvious approach to solve this problem would be to start with any coloring of G and keep augmenting the set of colored edges until we reach an optimal solution. This strategy produces an algorithm with running time $O(|E|^2)$, which is not so efficient. We will improve this algorithm by augmenting several paths at each stage. We first establish a couple of results.

Theorem 2.2: Let s be the maximum number of edges which can be colored with k colors in the bipartite graph $G = (V, E)$. Let C be a coloring such that $m = s - |E_c|$. Then there exist m edges disjoint color augmenting paths with respect to C, whose augmentation produces an optimal k-coloring of G.

Proof: By induction on m.

It is clear that the theorem holds for $m = 1$.

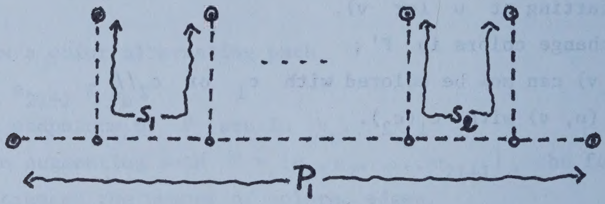
Assume that the result holds for all bipartite graphs and all colorings such that $m \leq m'$. We prove that the same holds true for $m' + 1$.

Let $G = (V, E)$ be a bipartite graph with a coloring C such that $|OPT| - |E_c| = m' + 1$. Let P_1 be an augmenting path with respect to C. Let C' be the k-coloring of G by augmenting P_1. Since $|E_{c'}| = |OPT| - m$, then, by the induction hypothesis, $E_{c'}$ contains m edge-disjoint augmenting paths with respect to C'. Let Q be such set. We will try to construct a set of $m' + 1$ edge-disjoint augmenting paths from P_1 and Q.

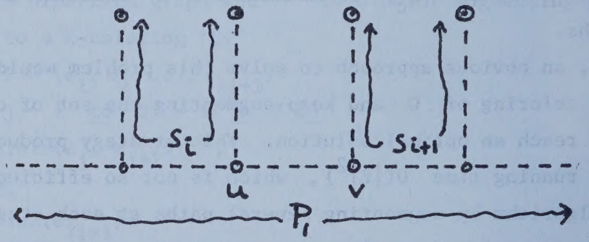
If P_1 is edge-disjoint from the paths in Q, then the theorem is true. Thus, assume that a set $S = \{s_1, \ldots, s_\ell\} \subseteq Q$ of paths have edges in common with P_1.

Claim: $S \cup \{P_1\}$ contains $\ell + 1$ edge-disjoint augmenting paths with respect to C.

Proof of the claim: We have the situation shown in the figure below.

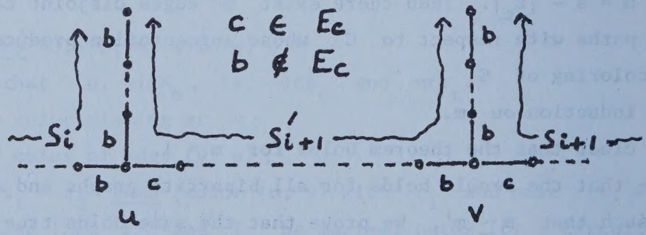


Consider s_i, s_{i+1} and P_1 for any $1 \le i \le \ell$. Let u and v be the vertices shown in the figure below.



Two cases have to be considered.

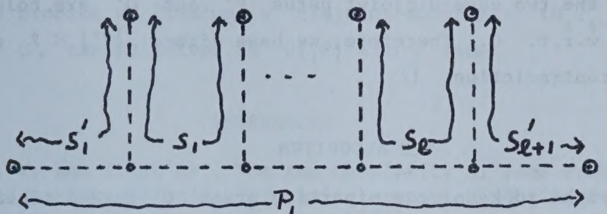
Case 1: The number of edges of P_1 between u and v is even. Then we can easily form a new augmenting path with respect to C as shown below



It is straightforward to check that s'_{i+1} is indeed an augmenting path.

Case 2: The number of edges of P_1 between u and v is odd. This case can be handled similarly.

Observe that we can set augmenting paths s_1' and $s_{\ell+1}'$ as shown below.



Hence $S' = \{s_1', \ldots, s_{\ell+1}'\}$ is a set of $\ell + 1$ edge-disjoint augmenting paths. This concludes the proof of the claim and the theorem. □

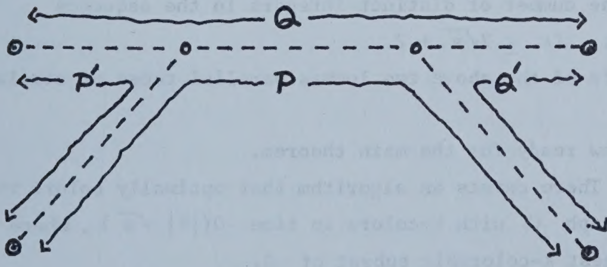
Unfortunately, the proof of the above theorem is not constructive. However, the following strategy looks promising; start with a coloring C. At each stage, find a maximal set of edge-disjoint color augmenting paths of shortest length. Repeat the procedure until no color augmenting paths exist. The key observation which makes the above strategy work is stated in the following theorem.

Theorem 2.3: Let C be a coloring (with k colors) of some of the edges of a bipartite graph G. Let ℓ be the length of a shortest augmenting path with respect to C and let S be a maximal set of edge-disjoint (color) augmenting paths of length ℓ. After augmenting S, G contains no augmenting paths of length $\leq \ell$.

Proof: Suppose G contains an augmenting path P of length $\leq \ell$, after being augmented by S. We distinguish between two cases.

(a) P and the augmenting paths in S are disjoint. This case can be handled easily.

(b) P has some edges in common with some members of S. Let Q be a member of S with edges common with P as shown in the figure below.



Using arguments similar to those in the proof of theorem 2.2, one can show that the two edge-disjoint paths P' and Q' are color augmenting paths w.r.t. C. Therefore, we have either $|P'| < \ell$ or $|Q'| < \ell$, a contradiction. □

THE ALGORITHM

Our algorithm to k-color a bipartite graph G is described in the following procedure.

```
procedure (G, K)
1.  while G contains a color augmenting path do
2.      $\ell \leftarrow$ length of shortest augmenting path;
3.      Let S be a maximal edge-disjoint set of augmenting
        paths of length $\ell$;
4.      Augment paths in S;
    endwhile
end
```

It is clear that the above algorithm finds the maximum number of edges which can be colored with k colors. On the other hand, it is not hard to implement steps 2 and 3 in $O(|E|)$ time. It remains to study the number of iterations of the while loop. The following two lemmas establish a bound on the number of iterations.

Lemma 3.1: Let s be the number of edges in an optimal k-coloring of G. Let C be a k-coloring of G with $|E_c| = r$. Then, if $s > r$, there exists a color augmenting path P with respect to C such that $|P| \leq 2\lfloor r/(s-r)\rfloor + 1$.

Let $P_0, P_1, P_2, \ldots$ be the set of all augmenting paths used by the above procedure.

Lemma 3.2: The number of distinct integers in the sequence $|P_0|, |P_1|, \ldots$ is $\leq 2\sqrt{s} + 2$.

The proofs of the above two lemmas parallel those of similar results in [HK].

We are now ready for the main theorem.

Theorem 3.3: There exists an algorithm that optimally colors the edges of a bipartite graph G with k-colors in time $O(|E| \sqrt{s})$, where s is the size of a largest k-colorable subset of G.

Corollary: Given a bipartite graph $G = (V, E)$ and positive integers k and α, the problem of finding k disjoint matchings $\{M_i\}_{i=1}^{K}$, such that $|M_i| \geq \alpha$, can be solved in $O(|E| \sqrt{K\alpha})$ time.

REFERENCES

[AJ] Ashok, V. and J. Ja'Ja', "On the Complexity of Some Problems Related to Edge Coloring," Technical report CS-80-10, Penn State University, University Park, PA 16802.

[E] Even, S., Graph Algorithms, Computer Science Press, 1979.

[GJ] Garey, M. and D. Johnson, Computers and Intractability, Freeman and Co., San Francisco, CA 1979.

[GK] Gabow, H. and O. Kariv, "Algorithms for Edge Coloring Bipartite Graphs and Multigraphs," private communication.

[HK] Hopcroft, J. and R. Karp, "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs," SIAM J. COMP. 2, 4, Dec. 1973, pp. 225-231.

[L] Lawler, E., Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, New York, 1976.

BANDWIDTH PROBLEMS IN GRAPHS

BURKHARD MONIEN
Fachbereich Mathematik-Informatik
Universität Paderborn
4790 Paderborn, West Germany

IVAN HAL SUDBOROUGH[†,*]
Electrical Engineering and Computer Science Department
Northwestern University
Evanston, Illinois 60201

ABSTRACT

The bandwidth$\overset{?}{\le}$k problem for an undirected graph G=(V,E) and a fixed integer k (denoted by k-BMP) asks whether a one-to-one function $f:V\to\mathbb{N}$, mapping the vertices to the natural numbers, exists such that every pair of adjacent vertices are mapped to integers at most k apart. It is shown that, for each $k\ge 1$, k-BMP can be solved nondeterministically in logarithm space and deterministically in time $O(n^k)$, where n is the number of vertices in G. This improves on an earlier result of Saxe.

The graph accessibility problem (GAP) is the problem of deciding whether a path exists in a finite directed graph from a specified start node to a specified goal node. It is known that the family of problems $\{GAP(f^k(n))\}_{k\ge 1}$, for appropriate functions f in the range $\log n\le f(n)\le n$, is log space complete for $\mathbb{N}SPACE(\log f(n))$, where GAP(f(n)) denotes the problem GAP restricted to graphs of bandwidth f(n). It is shown that $GAP(f(n))\in \mathbb{D}TISP(poly,f^2(n))$, where $\mathbb{D}TISP(poly,f(n))$ denotes the family of sets recognizable in polynomial time and simultaneous f(n) space deterministically. In particular, it follows that the family $\{GAP(\log^k n)\}_k\subseteq SC$, where $SC=\bigcup_k \mathbb{D}TISP(poly,\log^k n)$, and, hence, by the completeness result, $\mathbb{N}SPACE(\log\log n)\subseteq SC$

I. INTRODUCTION

Let G=(V,E) be a finite undirected graph. A *partial layout* of G is a one-to-one function f from a subset V_1 of V onto $\{1,\ldots,|V_1|\}$. The *bandwidth of a partial layout f* is the maximum difference between integers assigned to any pair of adjacent vertices, *i.e.* $\max\{|f(x)-f(y)| \mid x,y \in domain(f)\}$. A **partial layout** f is a *layout of G* if f is defined on all the vertices of G, *i.e.* domain(f)=V. The *bandwidth of G* is the minimum bandwidth of any layout of G, *i.e.* bandwidth(G)=min{bandwidth(f) | f is a layout}.

Bandwidth Minimization Problem: Given an arbitrary graph G and a positive integer k determine whether bandwidth(G)$\le$k.

---

† The work of this author was supported in part by NSF grant #MCS-79-08919

* work done during leave of absence to Fachbereich Mathematik-Info., Paderborn

In the bandwidth minimization problem both the graph G and the integer k are given as input values. A family of related problems is obtained by fixing the integer k at various values. For each integer $k \geq 1$, define:

Bandwidth$\overset{?}{\leq}$k problem: Given a graph G determine whether bandwidth(G) $\leq$k.

Papadimitriou has shown that the bandwidth minimization problem is $\mathbb{N}$P-complete.[5] Garey, Graham, Johnson, and Knuth have shown that this problem remains $\mathbb{N}$P-complete even for trees with no vertex of degree greater then three and that the bandwidth$\overset{?}{\leq}$2 problem can be solved in linear time.[3] Saxe extended this work by showing that the bandwidth$\overset{?}{\leq}$k problem, for each integer k, can be solved in $O(n^{k+1})$ steps.[7] Saxe also conjectured that one could improve on the complexity bounds given by his algorithm. We show that, for each k, the bandwidth$\overset{?}{\leq}$k problem can be solved in space log n nondeterministically and in time $O(n^k)$ deterministically. The algorithm presented shows, in fact, that the bandwidth$\overset{?}{\leq}$k problem, for each k, can be solved by a deterministic two-way pushdown automaton with k heads. The stated time bound then follows from the classical result of Cook: All k-head deterministic two-way PDA languages are recognized in $O(n^k)$ time on a random access machine.[1]

The following problem was originally described by Savitch:[6]

Graph accessibility problem (GAP): Given a finite directed graph G and two designated vertices s and t, determine whether there is a path from s to t.

The graph accessibility problem restricted to graphs of bounded bandwidth has also been previously considered by the authors.[4] Let GAP(f(n)) denote the **above** graph accessibility problem restricted to graphs $G=(V,E)$ of bandwidth $f(|V|)$, for any function f. (In this case the bandwidth refers to the bandwidth of the layout implicit in the encoding of the graph.) It is known that $\{\mathrm{GAP}(f^k(n))\}_k$ is log space complete for $\mathbb{N}$SPACE(log f(n)) and that $\mathrm{GAP}(f(n)) \in \mathbb{D}\mathrm{SPACE}(\log f(n) \times \log n)$, for appropriate functions f in the range $\log n \leq f(n) \leq n$.[4] In particular, it follows that $\mathbb{N}\mathrm{SPACE}(\log\log n) \subseteq \mathbb{D}\mathrm{SPACE}(\log n \log\log n)$ and $\mathbb{N}\mathrm{SPACE}(\log n) \subseteq \mathbb{D}\mathrm{SPACE}(\log^2 n)$. The algorithm cited for GAP(f(n)) takes time $f(n)^{k \log n}$, for some $k \geq 1$, and hence is not a polynomial time algorithm whenever f(n) is not a constant function. An algorithm is described here that shows that GAP(f(n)) can be solved in polynomial time and simultaneous $f^2(n)$ space. Cook has conjectured that GAP is not in the class *SC*, where *SC* denotes the family $\bigcup_k \mathbb{D}\mathrm{TISP}(\mathrm{poly}, \log^k n)$.[2] Our result shows that $\mathrm{GAP}(\log^k n)$ is in *SC*, for every k, and that $\mathbb{N}\mathrm{SPACE}(\log\log n) \subseteq$ *SC*, since *SC* is closed under log space reductions.

II. BANDWIDTH$\overset{?}{\leq}$k PROBLEM

Definition: Let $G=(V,E)$ be a finite undirected graph and k a positive integer. Let f be a partial layout from a subset V_1 of M elements of V to the positive integers. If an edge $e=\{u,v\}\varepsilon E$ is such that $u\varepsilon$domain(f) and $v\not\varepsilon$domain(f), then e is *dangling from f* or, simply, *dangling* when f is understood. The sequence $(f^{-1}(\max(M-k+1,1)),\ldots,f^{-1}(M))$ is the *active region of size at most k in f* or, simply, the *active region* when k and f are understood. A node is *active* if it is in the active region; it is *inactive* if it is in domain(f) and is not active; it is *unassigned* if it is not in domain(f). A partial layout f is *plausible for bandwidth k* or, simply, *plausible* when k is understood, if (1) $\{u,v\}\varepsilon E$ and $u,v\varepsilon$domain(f) implies that $|f(u)-f(v)|\leq k$, and (2) only active nodes have dangling edges. An edge$\{u,v\}$ is *backward*, if u and v are in domain(f) and exactly one is active.

A graph has bandwidth k iff there exists a plausible partial layout f (for bandwidth k) whose domain is all the nodes of G. It can be seen that a graph has bandwidth k iff there exists a sequence of partial layouts f_k, f_{k+1}, ..., $f_{|V|}$ such that (1) for all j $(k\leq j<|V|)$, f_{j+1} is an extension of f_j, (2) for all j $(k\leq j\leq|V|)$, f_j is defined on exactly j elements, and (3) for all j $(k\leq j\leq|V|)$, only the active nodes of f_j have dangling edges. We can easily check that these conditions are sufficient to guarantee that f_j, for each j, is plausible. First any partial layout defined on exactly k vertices is plausible, since all the nodes in the domain of the layout are active. We show that if f_j is plausible, then f_{j+1} is plausible. Let $e=\{u,v\}\varepsilon E$ and $u,v\varepsilon$domain(f_{j+1}). Then either (1) e is dangling from f_j, or (2) $u,v\varepsilon$domain(f_j). If e is dangling from f_j, but is not dangling from f_{j+1}, as in case (1), then e connects one of the active nodes u in f_j with the node v assigned to the integer $j+1$ by f_{j+1}. Since the active nodes of f_j are assigned integers between $j-k+1$ and j, the difference $|f(u)-f(v)|$ is bounded above by k. On the other hand, if $u,v\varepsilon$domain(f_j), as in case (2), then $|f(u)-f(v)|\leq k$ follows from the fact that f_j is plausible.

It follows that an algorithm to determine if a graph G has bandwidth k can simply determine whether a sequence of partial layouts f_k, f_{k+1}, ..., f_n exists which satisfies properties (1) through (3), where n is the number of vertices in the graph. In order to produce a partial layout f_{j+1} that is an extension of a partial layout f_j, we need to be able to find an unassigned node. It is easy to find an unassigned node, if a list of all the unassigned nodes is maintained. However, this would require space proportional to the number of nodes and we wish to construct an algorithm that uses only an amount of space that grows with the logarithm of the size of the graph. We shall store instead the list of *active nodes* of each successive partial layout. Along with the list of active nodes we store a list of backward edges. The unassigned nodes can be deduced from these two lists. That is, all the nodes in the current connected component(*) of the graph G are connected to the active nodes; the inactive nodes are connected to the active nodes only by paths that involve backward edges; the unassigned nodes are connected by paths that do not include any backward edges. For convenience, a path not containing a backward edge is called a *forward path*. In other words, the unassigned nodes can be located by a forward path routine. A nondeterministic Turing machine to determine whether a given node is reachable from an active node by a forward path and which takes logarithm space is constructed by a straightforward modification of the nondeterministic log space Turing machine which solves GAP.[6] The new Turing machine only needs to record the positions of the active nodes in the worktape space and, for each such active node, store in the finite state control the information about which edges attached to that node are backward. If the graph has bandwidth k, then there can be at most $\binom{k}{2}$ edges attached to any node and, hence, the list of backward edges is possible to store in the finite state control. Thus, we can show:

Theorem 1: For every $k \geq 1$, $k\text{-BMP} \in \text{NSPACE}(\log n)$.

Let us re-examine the problem of locating an unassigned node from a list of the active nodes and backward edges. Assume that $x_1, x_2, \ldots, x_k$ are the active nodes in a partial layout f_j defined on j elements, where $x_i = f^{-1}(j-k+i)$, for all i ($1 \leq i \leq k$). If there is an edge dangling from x_1, say the edge $\{x_1, y\}$, then the node y *must* be assigned to the integer j+1, if the extended layout is to be plausible. Otherwise, the node x_1 would be a node not in the active region of the extended partial layout with a dangling edge. On the other hand, if x_1 does not have a dangling edge in f_j, then it is unnecessary to store the node x_1 in a forward path routine. This is so, since the only edges attached to x_1 are either backward edges or are connected also to other active nodes. It follows that any forward path from node x_1 will also be a forward path from another one of the active nodes. So, either the unassigned node to be chosen for the extended partial layout is forced by an edge dangling from node x_1 or the node x_1 can be deleted from the list (leaving only k-1 nodes in a list of active nodes).

We construct an algorithm for k-BMP using a two-way deterministic pushdown automaton with k heads. The k heads are used, in general, to record the position of the active nodes in the graph. That is, the i-th

---

(*) We assume here, in fact, that G is connected. See Appendix.

input head is placed on that portion of the input string corresponding to the node x_i in the encoding of the graph. The pushdown store is used to record all of the nodes in the domain of the current partial layout. This is used to backtrack and try another possible partial layout , if a particular choice of assigning nodes does not give bandwidth k. The pushdown store enumerates these nodes in their assigned order. That is, the ℓ-th node from the bottom of the store is the node assigned to the integer ℓ in the current partial layout. If the current partial layout is defined on j nodes, then there will be j nodes stored in the pushdown store. Let us assume that at the moment the pushdown store represents some layout of j vertices and that the k heads are stationed to represent the nodes x_1, x_2, ..., x_k in the active region.

If the node x_1 is attached to a dangling edge, say the edge $\{x_1,y\}$, then the PDA can simply move the first input head from the position for x_1 to the position for y. This is possible, since the edge $\{x_1,y\}$ is included in the input string at the position corresponding to node x_1 in the encoding of G. If the node x_1 does not have a dangling edge, then head one becomes free and the PDA can use this head to locate an unassigned node to be used as a candidate for the assignment to the next integer j+1. The PDA will try each successive node in the graph, in the order listed in the input string encoding of G, as a candidate for the next unassigned node. In each case the PDA must decide whether the candidate node is an unassigned node or not. If the candidate node is an assigned node, then the PDA must continue by trying the next node as candidate. Again, a candidate node is unassigned iff there is a forward path to the node from an active node. We construct a PDA that will determine whether the candidate node is reachable by a forward path and simultaneously possible to assign to the integer j+1 in some plausible partial extension of f_j. A possible situation is illustrated in Figure 1 below for bandwidth four. We observe that a forward path does exist in the graph from x_2 to the candidate node y. The forward path can be detected by a four head PDA as follows: It places its four heads on nodes y,z_1,z_2, and z_3. (The first head is placed on node y at some point in its enumeration of all the possible candidate nodes; the other nodes are located from their attached edges to x_2,x_3, and x_4.) The PDA observes that no edge connects the vertices its heads are

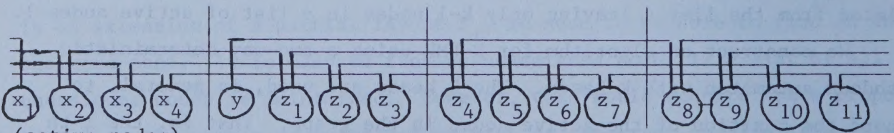


Figure 1

scanning. (If there were an edge connecting y with one of the nodes z_1,z_2, or z_3, then the PDA would know that there is a forward path from an active node to the candidate node y.) The PDA moves its four heads to the nodes z_4,z_5,z_6, and z_7. Again, in this case, no edge connects any of these vertices. So, the PDA moves its four heads to vertices z_8,z_9,z_{10}, and z_{11}. Since, in this case there is a connection between z_8 and z_9, the PDA knows that node y is connected by a forward path to one of the active nodes. The algorithm can be thought of as one that moves the four heads from one block of four nodes to the next block of four nodes. It keeps track of which of the nodes in the block are connected to node y and which of the nodes in the block are connected to the active nodes. When there is an edge connecting these two parts of a block, then the PDA knows that there is a forward path to the node y. If the number of nodes in such a block ever becomes larger than four, then the PDA knows that either there does not exist a forward path to node y from the active nodes or that y cannot be assigned to the integer j+1 in any plausible extension of the partial layout f_j.

This process can be described in a more formal way as follows: Let the current candidate node be node y. First, the PDA verifies that y is not one of the active nodes $x_1,x_2,\ldots,x_k$ and, furthermore, is not connected to any of these nodes by a backward edge. It follows, therefore, that if y is an assigned node, then the shortest path between y and an active node is of length at least two. Let X denote the set of nodes connected by a dangling edge to an active node. If $y \in X$, then the PDA knows that y is an unassigned node and, hence, is an appropriate candidate. If $y \notin X$, then there can be at most k-1 nodes in X, if y is a node that can possibly be assigned to the integer j+1 in a plausible extension of the partial layout f_j. That is, all of the nodes in X must be assigned to integers between j+2 and j+k, if y is assigned to j+1 and $y \notin X$. So, if there are more than k-1 nodes in the set X, then the PDA knows that y is not an appropriate candidate and tries the next candidate for a node to assign to the integer j+1. On the other hand, if X contains at most k-1 nodes, then the PDA can position k-1 of its heads on these nodes and position the remaining head on node y. The PDA now checks to see whether there is an edge connecting node y to one of the nodes in the set X. If so, then y is an unassigned node, since a forward path exists, and the PDA has verified that y is a good candidate. We may observe that, if y is an assigned node, then the shortest path from one of the nodes in X to the node y must be of length at least three. This is true, since all nodes in the set X are one step further away from y than the active nodes, if y is an assigned node.

Let f be a tentative partial layout, which extends the partial layout f_j, such that y is assigned the integer j+1 and the nodes in X are assigned to integers between j+2 and j+k. We note that f may not be a partial layout since y may be a previously assigned node. In the tentative layout f, node y can have only dangling edges, since if it had an edge connected to a node in X or an active node, then we would have already discovered that y is an unassigned node. Let Y={y}. Define X' and Y' to be the sets of nodes connected by dangling edges to X and Y, respectively. If $X' \cap Y' \neq \Phi$ or if there is an edge connecting a node in X' to a node in Y', then node y is an unassigned node, since we have verified the existence of a forward path. If $X' \cup Y'$ contains more than k nodes, then the node y is an assigned node or is not possible to assign to j+1 in any plausible partial layout. In this case the PDA would continue by trying the next possible candidate for this assignment. So, we assume that $X' \cup Y'$ contains at most k elements. Then the k heads of the PDA can be shifted to the k nodes in this set. If either X' or Y' is empty, then the PDA need not continue with the verification of y, since the node y cannot be reached from an active node by a forward path. We observe also that, if y is an assigned node, then the shortest path from a node in X' to a node in Y' must be of length at least three. This is so, since in moving from X to X' we move one step further away from the nodes in Y and in moving from Y to Y' we can move at most one step closer to nodes in X. Therefore, there is a path of length at most one connecting a node in X' to a node in Y' only if y is an unassigned node. This process is now repeated by setting X to X' and Y to Y' and following the steps outlined above. Eventually, in repeating these steps, if y is an unassigned node that can be assigned to the integer j+1 in a plausible extension of f_j, the PDA will find a path from X' to Y' of length 1 or less. The verification process for the node y then terminates successfully. The node y is known to be an unassigned node. This verification process uses k heads. In this way we construct a k-head deterministic two-way pushdown automaton to determine whether a graph has bandwidth k by trying all possible sequences of partial layouts via backtracking, whenever required, using the information stored in the pushdown store. By this technique we show:

Theorem 2. For every $k \geq 1$, k-BMP can be solved by a k-head deterministic two-way pushdown automaton.

Corollary: For every $k \geq 1$, k-BMP can be solved by a random access machine in $O(n^k)$ steps.

III. $GAP(f(n)) \in \mathbb{D}TISP(poly,f(n))$

Let $G=(V,E)$ be a finite directed graph and s and t be two vertices. G is presented as an input string and, hence, there is an implicit layout, called the *input layout*, of G given to us. We check that the input layout has bandwidth $f(|V|)$, where f is a specified constructible function on the natural numbers. That is, we check that each edge $\{i,j\}$ is such that $|i-j| \leq f(|V|)$. (Note that we refer to the nodes as the i-th node and the j-th node using the ordering given by the input layout.) This verification can clearly be done in $f(|V|)$ space and polynomial time. Therefore, we proceed to verify that the graph G possesses a path from s to t. This is done by computing information sequentially about sets of consecutive nodes of the same cardinality as the bandwidth of G. These sets are called windows.

The *window of size k starting at position m*, denoted by $W(k,m)$, is the set of nodes $\{ x \in V \mid m \leq x \leq m+k-1 \}$. With each position m ($1 \leq m \leq |V|-k+1$) we associate a 0-1 valued $k \times k$ *connection matrix* C_m such that $C_m(i,j)=1$ iff there is a path in G from the i-th node in the window $W(k,m)$ to the j-th node in the window $W(k,m)$ which passes only through nodes numbered at most $m+k-1$. The matrix C_1 is straightforward to compute from the edges connecting the first k nodes of G. In the general case, C_{i+1} can be computed from just the matrix C_i and the edges entering into or leaving from node $i+k$, *i.e.* the last node of the window $W(k,i+1)$. In particular, C_{i+1} is computable in k^2 space and time polynomial in k from the previous matrix C_i.

In addition, we define, for each position m, two 0-1 valued k-tuples R^s_m and R^t_m such that: (1) $R^s_m(i)=1$ iff there is a path from node s to the i-th node of the window $W(k,m)$ which does not pass through any node higher than $m+k-1$, and (2) $R^t_m(i)=1$ iff there is a path from the i-th node of the window $W(k,m)$ to node t that does not pass through any node higher than $m+k-1$. It is straightforward to compute R^s_1 and R^t_1 by looking at the first k nodes of G. It is also straightforward to compute R^s_{i+1} and R^t_{i+1} from R^s_i and R^t_i and the connection matrix C_{i+1}. Each such computation takes space at most k^2 and time polynomial in k. It can be seen that there is a path from s to t iff for some m ($1 \leq m \leq n-k+1$) and some i ($1 \leq i \leq k$) $R^s_m(i)=1$ and $R^t_m(i)=1$. Let $G=(V,E)$ be a graph with bandwidth $f(|V|)$. Then, the windows and the vectors R^s_m and R^t_m are of size $f(|V|)$ and the connection matrix is of size $f^2(|V|)$. It follows that a Turing machine can be constructed to solve $GAP(f(n))$ in space $f^2(n)$ and time $n \times f^c(n)$, for some $c>1$. Therefore, we can show:

Theorem 3: For each constructible function f such that $f(n) \leq n$, $GAP(f(n)) \in \mathbb{D}TISP(poly,f^2(n))$.

In particular, for any $k \geq 1$, $GAP(\log^k n) \in \mathbb{D}TISP(poly,\log^{2k} n)$ and, therefore, $\{GAP(\log^k n)\}_k \subseteq SC$. The next corollary follows from the completeness of $\{GAP(\log^k n)\}_k$ for NSPACE(loglog n) and Theorem 3:

Corollary: NSPACE(loglog n) $\subseteq$ *SC*.

APPENDIX

If a graph G=(V,E) is not connected we can check that G has bandwidth k by laying out each connected component separately. G has bandwidth k iff every component has bandwidth k. A nondeterministic log n space bounded Turing machine may not be capable of identifying that two nodes lie in separate components. Therefore, we iteratively lay out the component containing node number 1, then the component containing node number 2, and so on. This will, of course, involve duplication of effort for components that contain more than one node, but it will ensure that all components will be checked for bandwidth k. Furthermore, for each node i, we obtain an initial partial layout of the connected component containing node i by guessing k nodes one at a time and verifying that each is connected to node i using a path routine. (We may also guess nondeterministically that the component containing node i has fewer than k nodes and verify that this is true, too. In this case, the component *must* have bandwidth k, since it has fewer than k vertices.) Therefore, it is really sufficient to build a nondeterministic log n space bounded Turing machine to check for bandwidth k in connected graphs as we have done in section two.

LITERATURE

(1) Aho, A.V., J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Massachusetts (1974).

(2) Cook, S.A., Towards a complexity theory of synchronous parallel computation, Technical Report #141/80, University of Toronto, CANADA

(3) Garey, M.R., R.L. Graham, D.S. Johnson, and D.E. Knuth, Complexity results for bandwidth minimization, *SIAM J. on Applied Mathematics* 34 (1978), pp. 477-495.

(4) Monien, B. and I.H. Sudborough, Eliminating nondeterminism from Turing machines which use less than logarithmic worktape space, 1979 ICALP Proceedings, Vol. 72 *Lecture Notes in Computer Science*, Springer Verlag, pp. 431-445.

(5) Papadimitriou, C.H., The NP-completeness of the bandwidth minimization problem, *Computing* 16 (1976), pp. 177-192.

(6) Savitch, W.J., Relationships between nondeterministic and deterministic tape complexities, *J. Computer and System Sci.* (1970), pp. 177-192.

(7) Saxe, J.B., Dynamic programming algorithms for recognizing small bandwidth graphs in polynomial time, Technical Report, Computer Science Department, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 15213, U.S.A.

ANALYSIS OF A CLASS OF GRAPH PARTITIONING PROBLEMS

P. BERTOLAZZI, M. LUCERTINI, A. MARCHETTI SPACCAMELA
Centro di Studio dei Sistemi di Controllo e Calcolo Automatici del
CNR e Istituto di Automatica dell'Università di Roma, Via Eudossiana 18
00184 - ROMA ITALY

ABSTRACT

The partitioning problem of a graph into m subgraphs with size constraint and an objective function given by the sum of links among vertices belonging to different subgraphs, is well known and appears in a number of very diverse problem areas.

This paper presents some simple results concerning some particular classes of trees such as chains, cycles, stars and binary trees. Furthermore some ideas for further research in the more general graph partitioning problem are presented.

1. INTRODUCTION

Given a graph $G(V,E)$ with $|V| = n$ and $|E| = p$, weights $w(v) \in Z^+$ for each $v \in V$ and $a(e) \in Z^+$ for each $e \in E$, and a positive integer B, the graph partitioning problem (GP) is the problem of finding a partition of V into disjoint sets $\{V_1, V_2, \ldots, V_m\}$ such that $(\sum_{(v \in V_i)} w(v) \leq B)$ for $(1 \leq i \leq m)$ and such that if $E^c \subseteq E$ is the set of edges that have their two end points in two different sets V_i, then $(\sum_{(e \in E^c)} a(e))$ is minimal.

This problem is relevant in many fields and in particular it frequently arises in computer system design and in the allocation of computer information to blocks of storage [1,2,3,4].

The (GP) problem has been proved to be NP-complete for fixed $B \geq 3$, even if all vertex and edge weights are equal to 1 [5,6]. It remains NP-complete with the additional constraint that the graph obtained from

---

This work was supported by Consiglio Nazionale delle Ricerche.

the partition is acyclic [5]. If G is a tree the general problem is NP-complete, but can be solved in pseudopolynomial time $o(n \cdot B^2)$ [7]; if all edge weights or vertex weights are equal the tree problem can be solved in polynomial time [5,8].

In this paper some simple additional results concerning some particular classes of trees are showed and some ideas for further research in the more general case are presented.

In particular the NP-completeness of simple structures such as stars and binary trees is proved, an $o(n^2)$ algorithm for the "chain partitioning" problem (CP) and an $o(n \cdot B)$ algorithm for the "star partitioning" problem (SP) are presented. The (CP) algorithm is generalized in order to find an heuristic to solve the general (GP) problem.

2. NP-COMPLETENESS OF SOME SIMPLE CLASSES OF TREES

THEOREM 1. The problem of finding an optimal partitioning of a star is NP complete.

PROOF. We show that the subset sum problem (shown to be NP complete by Karp 72) is polynomially reducible to our problem. Given a set $N = \{a_1, a_2, \ldots, a_n\}$ such that

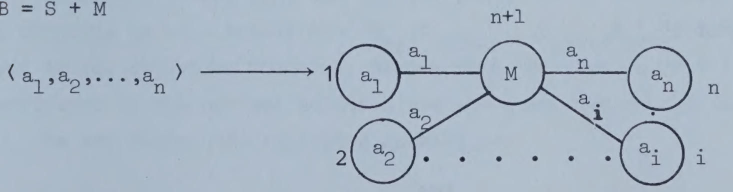
$$\sum_{i=1}^{n} a_i = 2S$$

of positive numbers, does exist a subset $J \subseteq N$ such that

$$\sum_{j \in J} a_j = \sum_{j \notin J} a_j = S?$$

Given any istance of subset sum we can construct an istance of the partitioning problem for stars in the following way:

- for each $i = 1,2,\ldots,n$ there is a node i with weight a_i;
- there is an additional node $(n+1)$ with weight $M = 2S$;
- for each $i = 1,2,\ldots,n$ there is an edge e_i with weight a_i between the node i and the node $(n+1)$;
- $B = S + M$



Now we show that the subset sum has solution if and only if the corresponding istance of the graph partitioning problem has a solution less or equal to S.

A) ⇒ Let J be a solution for the subset problem. We can obtain a solution for the GP problem by joining node (n+1) with the nodes j, $j \in J$. It is easy to see that the capacity constraint is satisfied and the value of the solution is S.

B) ⇐ Let it be given a solution of the GP problem whose value is less or equal to S and let I be the set of nodes in the same partition of node (n+1).

For the capacity constraint of the bin we have

$$\sum_{i \in I} a_i + M \leq S+M \Rightarrow \sum_{i \in I} a_i \leq S \tag{1}$$

Furthermore as the value of the solution is less or equal to S we have

$$\sum_{i \notin I} a_i \leq S \tag{2}$$

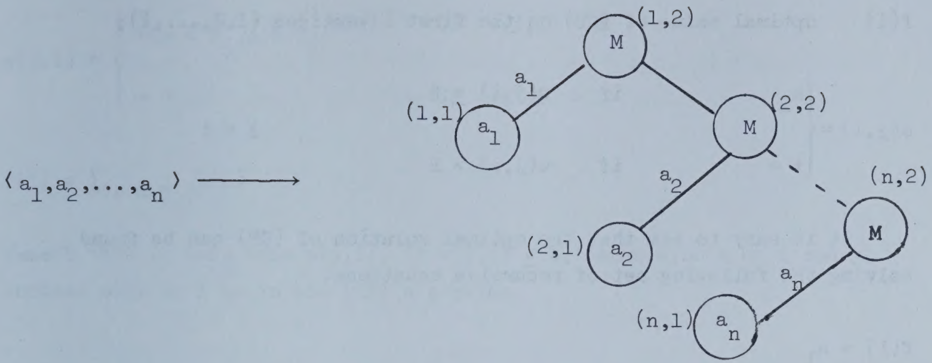
By considering (1) and (2) we have $\sum_{i \in I} a_i = S = \sum_{i \notin I} a_i$. This completes the proof. ◀

With a slight modification of the proof we can obtain the following corollary.

THEOREM 2. The problem of finding an optimal partition of a binary tree is NP-complete.

PROOF. Given an istance of subset sum we can construct an instance of GP problem for binary tree in the following way:

- for each i = 1,2,...,n there are two node (i,1) and (i,2) with weight a_i and M = 2S respectively;
- foe each i = 1,2,...,n there is an edge e_i between (i,1) and (i,2) with weight a_i;
- for each i = 1,2,...,n-1 there is an edge e_{n+1} between (i,2) and ((i+1),2) with weight 2S;
- B = (2n+1)S = nM + 1



We show that the subset sum problem has solution if and only if the graph partitioning problem has solution with value less or equal to S.

A) ⇒ If J is a solution of the subset sum problem than we obtain a feasible solution for the graph partioning problem joining the nodes (j,2) j=1,2,....,n and the nodes (j,1), $j \in J$. The cost of this solution is S.

B) ⇐ If there is a solution with value less or equal to S then all the nodes (i,2) must be in the same cluster with some other nodes (j,1) for $j \in J$. It is not difficult to see that $\sum_{j \in J} a_j \leq S$ (for the capacity constraint) and that $\sum_{j \notin J} a_j \leq S$ (because the solution is less a equal to S). Therefore $\sum_{j \in J} a_j = S$ and so there is a solution for the subset sum problem.◀

3. A $o(n^2)$ ALGORITHM FOR (CP)

Let $1, e_{12}, 2, e_{23} \ldots, n-1, e_{n-1n}, n$ be a sequence of nodes and edges: this sequence is called a chain leading from 1 to n. The (CP) problem is the (GP) problem on a chain, where w_i is the weight of vertex i and a_i is the weight of the edge e joining i to i+1.

Let $\Omega_{ji}: \{j+1,\ldots,i-1,i\}$ be a subset of the nodes of the chain. Let be $\Omega = \{\Omega_{ji}\}$ $\forall j, \forall i > j$; it is easy to see that $|\Omega| = n \cdot (n+1)/2$. Let's associate a weight to each subset Ω_{ji}, $w(j,i) = \sum_{k=j+1}^{i} w_k$: the set $\Omega' = \{\Omega_{ji}: w(j,i) \leq B\} \subset \Omega$ has cardinality $|\Omega'| < n \cdot (n+1)/2$.

It's easy to see that the optimal partition of the chain is obtained by choosing in Ω' a subset $P = \{\Omega_{oj}, \Omega_{j+1k}, \ldots, \Omega_{m+1\ell}, \Omega_{\ell+1n}\}$ with minimum cost of the objective function. Remark that the subsets of V that are candidates to the optimal solution are all connected subchains.

We can define the following quantities:

f(i) optimal value of (CP) on the first i vertices (1,2,...,i);

$$\varphi(j,i) = \begin{cases} a_i & \text{if} \quad w(j,i) \leq B \\ & \qquad\qquad\qquad\qquad j < i \\ +\infty & \text{if} \quad w(j,i) > B \end{cases}$$

It is easy to see that the optimal solution of (CP) can be found solving the following set of recursive equations.

$$f(1) = a_1$$

$$f(i) = \min_{j<i}(f(j) + \varphi(j,i)) \qquad (i = 2,3,\dots,n)$$

In the worst case the complexity of this algorithm is obviously $o(n^2)$. The problem remain $o(n^2)$ even if the constraints on the maximum size of the subsets are more complicate (for example if w_i is a r-vector and the constraint has the general form $g_k(w_i,\ i \in V_k) \leq 0$).

This algorithm can be utilized for solving the "cycle partitioning" problem (CLP),i.e. the (CP) problem with an additional edge between vertices 1 and n. Given an instance of the (CLP), let $\sum_{k=1}^{n} w_k > B$, in this case at least two cut will exist in the cycle; let c_i be the optimal value of the solution of the (CP) problem obtained from the cycle dropping the edge (i,i+1). The optimal solution of (CLP) problem is given by $\min_i(c_i + a_i)$. If we start from vertex 1, the procedure can stop when i = j with j = $= \min\{k | \sum_{i=1}^{k} a_i > B\}$. In the worst case the complexity is $o(n^3)$.

4. A GENERALIZATION OF THE (CP) ALGORITHM

Let us define a new problem: the graph partitioning problem (GP) with the additional constraint that, given an ordered list of vertices $L = \{r_1, r_2, \dots, r_n\}$ each subset of the optimal partition only contains adjacent nodes of L. Without loss of generality we can suppose that the vertices are renumbered such that L = {1,2,...,n}; let T_i be the set of edges belonging to the cut {(1;2,...,i), (i+1,...,n)} (i.e.edges e_{hk}) such that h ∈ (1,2,...,i) and k ∈ (i+1,...,h)). The set of recursive equations of the previous section holds to find the optimal solution, if we define φ(j,i) and f(1) as follows:

$$\varphi(j,i) = \begin{cases} \sum_{(e \in T_i, e \notin T_j)} a(e) & \text{if } w(j,i) \leq B \\ +\infty & \text{if } w(j,i) > B \end{cases} \qquad j < i$$

$$f(1) = \sum_{(e \in T_1)} a(e)$$

Remark that in this case $\varphi(j,i)$, if $w(j,i) \leq B$, depends both on i and j, instead only on i as in the (CP) algorithm.

5. A o(n•B) ALGORITHM FOR (SP)

The (SP) problem is the (GP) problem on a tree, where one vertex (namely vertex 1) is connected with all other vertices. Let w_i be the weigh of vertex i and a_i be the weight of the edge between vertices 1 and i.

Let f(i,j) be defined as:

f(i,y) optimal solution of (SP) on the first i vertices with weight of node 1 equal to y;

the optimal solution of (SP) can be found solving the following set of recursive equations:

$$f(1,y) = 0 \qquad , \forall y$$

$$f(i;y) = +\infty \qquad , \forall i, \forall y > B$$

$$f(i,y) = \min\{a_i + f(i-1,y) \qquad , f(i-1,y+w_i)\}$$

$$i = 2,3,\ldots,n$$

$$y = a_1, a_1+1, \ldots, B$$

The algorithm finds the optimal solution in time o(n•B).

Remark that we can apply the same techniques used for the knapsack problem in order to obtain ε-approximate solutions in polynomial time [9].

6. FURTHER RESEARCH

Algorithms (axact or approximate) for solving (GP) can be based on the (CP) (or the (CLP)) algorithm, on the ground of the results of section 4.

Supposing that n_i be the number of vertices in the set i in the optimal solution and m be the number of sets, the problem became how to find one of the $((\prod_{i=1}^{m} n_i!)m!)$ optimal sequences of vertices among the $(n!=(\sum_{i=1}^{m} n_i)!)$ lists that we can generate. The approach can be in two different directions: the first one is to find a set of dominance relations among the sequences, in order to eliminate as many sequences as possible; the second one is, given a sequence and the associate optimal solution, to point out a heuristic in order to find better sequences.

An algorithm that seems to work well in many practical cases is the following. Let p(L) be the (GP) problem given the list L (see section 4) and z(L) the corresponding solution.

```
0. Solve p(L). $H = \emptyset$
1. Find (h,k) $(h < k)$ such that
   $a_{hk} = \max\{a(e): e \in \bigcup_i T_i, e \notin H\}$
   Solve p(L') with L' obtained from L making k adjacent to h and
   renumbering the vertices such that $k = h+1$ and $j = i+1$
   $(h < j < k)$ (if $k = h+1$ then go to 3).
2. If $z(L') < z(L)$ then set $L = L'$, update $\{T_i\}$, set $H = \emptyset$
   and go to 1 else increase $H = H \cup (h,k)$.
3. If $M = \bigcup_i T_i$ then stop, else go to 1.
```

Further theoretical results and computational experiences in this area will be presented in a forthcoming paper.

ACKNOWLEDGMENTS

The authors would like to thank G. Ausiello for many helpful comments during the development of this research.

REFERENCES

[1] CHRISTOFIDES N., P.BROOKER: *The optimal partitioning of graphs*, SIAM J. App. Math., Jan 1976.

[2] RAO G.S., H.S. STONE, T.C.HU: *Assignment of tasks in a distributed peocessor system with limited memory*, IEEE tr. on Comp., C-28, 1979.

[3] LAWLER E.L., K.N. LEVITT, J. TURNER: *Module Clustering to Minimize Delay in Digital Networks*, IEEE Tr. on Comp. C-18, 1969.

[4] KERNIGHAN B.W.: *Some graph partitioning problems related to program segmentation*, PhD Thesis, Princeton Univ., NI., Jan 1969.

[5] GAREY M.R., D.S. JOHNSON: *Computers and Intractability*, Freeman and Co., 1979.

[6] HYAFIL L., K.L. RIVEST: *Graph partitioning and constructing optimal decision trees are polynomial complete problems*, Re. 33 IRIA-Laboria, Rocquencourt, France.

[7] LUKES J.A.: *Efficient algorithm for the partitioning of trees*, IBM J. Res. Develop. 18, 1974.

[8] HADLOCK F.O.: *Minimum spanning forest of bounded trees*, Proc. 5th Southeastern Conf. on Combinatorics, Graph theory and Computing, Winnipeg., 1974.

[9] LAWLER E.L.: *Fast approximation algorithms for knapsack problems*, Proc. 18th Ann. Symp. on Foundations of Computer Science, IEEE Comp. Soc., Long Beach, 1977.

ALGORITHMS FOR FINDING PATHS WITH MULTIPLE CONSTRAINTS

JEFFREY M. JAFFE
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

ABSTRACT

Let $G = (V, E)$ be a graph with weight function $w{:}E \rightarrow \mathbb{Z}^+$ and length function $\ell{:}E \rightarrow \mathbb{Z}^+$. The problem of determining for $v_1, v_2 \in V$ whether there is a path from v_1 to v_2 with weight at most W and length at most L is NP-complete. This paper gives two approaches to meeting or approximating the length and weight constraints.

The first approach is to use a pseudo-polynomial time algorithm which determines whether a path meets the constraints. Its running time is $O(n^5 b \log nb)$ where $n = |V|$ and b is the largest length or weight. If tables with $O(n^3 b)$ entries are kept then *all* instances of multiple constraints may be decided.

In the second approach, an objective function is defined which evaluates a path's distance from meeting the constraints. Polynomial time algorithms attempt to find good paths in terms of the objective function. The best algorithm is at most 1.5 times worse than optimal when a feasible path exists.

1. INTRODUCTION

Distributed algorithms for the shortest path problem have attracted considerable interest recently [1]. Part of the reason for this interest is due to applications in message routing in computer networks [2]. If a message or a sequence of messages needs to be sent from one node to another, it is desirable to use the shortest route, least cost route, or shortest delay route—all variants of the shortest path problem.

To provide flexible service to network users, one might like to minimize some combination of several criterion, e.g. achieve relatively low delay at relatively low (dollar) cost. One possible criterion is to minimize some linear combination of delay and cost. This may be reformulated as another shortest path problem and may be easily solved in polynomial time. Another formulation is to insist that the chosen path be within a given cost and given delay. Determining whether such a "feasible" path exists is NP-complete [3], and it is this problem that we address herein.

In Section 2 a pseudo-polynomial time algorithm is given for the problem. The running times is $O(n^5 b \log nb)$ where n is the number of vertices or *nodes* and b is the largest cost or delay (henceforth called length and weight). When executed in a distributed manner, $O(n^4 b \log nb)$ machine steps are required per network node.

Time $O(n^5 b \log nb)$ may be too expensive for certain application. The rest of the paper describes polynomial time algorithms that *approximate* the constraints on the path. First one must define the notion of approximating the constraints. Section 3 defines an objective function used to evaluate a path's distance from the constraints.

In Section 4 we use shortest path algorithms to approximate constraints when a feasible path exists. An algorithm that uses very limited information is at most twice worse than optimal. A more complicated algorithm is at most 1.5 times worse than optimal. A measure is defined on the problem space, which enables us to discuss the average performance of the algorithm. The average for the first algorithm is $1 + \log 2 \approx 1.69$ and for the second is $3 - (\pi/2) \approx 1.43$.

2. A PSEUDO-POLYNOMIAL ALGORITHM

A *graph* $G = (V, E)$ consists of a set V of *vertices,* and a set E of *edges*. Associated with each edge $e \in E$ is a *length* $\ell(e) \in \mathbb{Z}^+$ and a *weight* $w(e) \in \mathbb{Z}^+$. A *path* p is a sequence of distinct vertices $v_1, ..., v_k$ such that $(v_i, v_{i+1}) \in E$ for $i=1, ..., k-1$. The set $\{(v_i, v_{i+1})\}$ is called the *edges of p.* The length (weight) of p, L(p) (W(p)) is the sum of the lengths (weights) of the edges of p. There is a polynomial time algorithm that given any graph, length function, and pair of vertices, finds the minimum length path between the vertices [4].

Consider the following decision problem:

MULTIPLE CONSTRAINTS on a PATH (MCP): Given a graph $G = (V, E)$, length and weight functions ℓ, w on G, integers L, W, and $v_1, v_2 \in V$; is there a path p from v_1 to v_2 such that $L(p) \leq L$ and $W(p) \leq W$?

This problem is NP-complete [3]. As explained in the introduction, this problem has a number of important applications, among them that of finding a route in a computer network with cost at most L and delay at most W. Our first approach to MCP, solves it in pseudo-polynomial time. If b is the largest length or weight in the network, the algorithm runs in time bounded by $O(n^5 b \log nb)$.

We informally review some definitions [3]. An algorithm runs in *polynomial time* if its run-time is polynomial in the *length* of the input. Typically, a number x is represented in log x bits, and an instance of MCP is of length between O(n) and O(n log b). An algorithm runs in *pseudo-polynomial* time if its run-time is polynomial in the length of the input *and largest number* in the input. The largest number is b. $O(n^5 b \log nb)$ is not a polynomial in O(n log b), but is in n and b. Thus the $O(n^5 b \log nb)$ algorithm is a pseudo-polynomial time algorithm.

A special case of this algorithm is noted in [3]. If all weights or all lengths are equal, then a polynomial time algorithm exists.

The algorithm is presented here as a "distributed" algorithm (where the vertices are thought of as nodes in a network) with $O(n^4 b \log nb)$ computation done at each node. Since there are n nodes, $O(n^5 b \log nb)$ time is required for a centralized computation.

We first outline the algorithm. Let b be the largest weight in the graph. All loop-free paths between any two nodes in the graph have weight at most nb. Also, any path has integral weight. The strategy is to simultaneously find the minimal length path between all pairs of nodes with weight at most 1, 2, ..., nb. Each node maintains a table of O(nb) entries for every other node representing the shortest lengths for weights 1, ..., nb. This requires a total of $O(n^2 b \log nb)$ storage at each node, and $O(n^3 b \log nb)$ in the entire network, since each entry is of size at most log(nb). (Each entry is of size log nb since it represents the length of a loop-free path).

Using the tables, it is easy to solve any instance of MCP. Given v_1, v_2, L, and W, one checks the table at v_1 for paths to v_2 with weight W, to find the minimal length path. If it exceeds L, then there is no feasible solution (i.e. no paths meet the constraints), otherwise there is a feasible path.

Initially the tables have little information. Let $L(v_1, v_2, k)$ be the entry in the table of v_1 which will ultimately contain the length of the shortest path to v_2 with weight at most k ($v_1 \neq v_2$). Then $L(v_1, v_2, k)$ is initialized according to information known locally to v_1. In particular, $L(v_1, v_2, k) = \infty$ if $(v_1, v_2) \notin E$ or if $e = (v_1, v_2) \in E$ but $w(e) > k$. If $e = (v_1, v_2) \in E$ and $w(e) \leq k$, then $L(v_1, v_2, k) = \ell(e)$. Thus $L(v_1, v_2, k)$ is the length of the "best known" path, of weight at most k. For convenience, we say that the initial values are the entries *after the first iteration*. With the above preliminaries, we give the algorithm to solved MCP:

Algorithm for node v_1 at ith iteration ($i \geq 2$).

```
1.  For each entry $L(v_1, v_2, k)$ that was changed at the i-1st iteration
        For each $v_3$ such that $e = (v_1, v_3) \in E$ with $w(e) = w$ send a message to $v_3$
        "$L(v_3, v_2, k+w) \leq L(v_1, v_2, k) + \ell(e)$".
2.  For each message received of the form "$L(v_1, v_2, k) \leq x$"
    a.  If $L(v_1, v_2, k)$ is already less than x ignore the message
    b.  If $L(v_1, v_2, k)$ is currently greater than x, then $L(v_1, v_2, k) := x$.
    c.  If $L(v_1, v_2, k') > x$ for $k' > k$, $L(v_1, v_2, k) := x$.
```

Basically, in step 1, node v_1 informs all neighbors about improved paths learned at the last iteration. In step 2, node v_1 learns about other improved paths from its neighbors.

There are certain assumptions made about the above algorithm. It is assumed that synchronization prevents a node from starting the i+1st iteration before all have completed the ith. The above algorithm has various inefficiencies which are removed later. For now the above simple form is used. The correctness of this algorithm is encapsulated in Theorem 1.

Theorem 1. After at most n−1 iterations of the above algorithm, each entry $L(v_1, v_2, k)$ for $v_1, v_2 \in V$, $1 \leq k \leq nb$ contains the length of the minimal length path from v_1 to v_2 with weight at most k.

It is left to the reader to verify that—if at any point in the algorithm $L(v_1, v_2, k) = L$, then there is a path of length L and weight at most k from v_1 to v_2.

The proof of Theorem 1 is facilitated with the following:

Lemma 1. Assume that a minimal length path from v_1 to v_2 with weight at most k traverses r edges. Then $L(v_1, v_2, k)$ contains the length of this path after the rth iteration.

Note that all loop-free paths require at most n−1 edges. Since table entries are never increased, Theorem 1 is immediate from Lemma 1, using the remark that preceded Lemma 1.

Proof of Lemma 1. Induction on r:

$r = 1$. Assume that the minimal length path of weight at most k consists of a single edge $e = (v_1, v_2)$. Then when initializing L, v_1 sets $L(v_1, v_2, k) := \ell(e)$. Thus the entry is correctly entered at the first iteration.

Inductive Step. Assume that the smallest (in terms of number of edges) minimal length path from v_1 to v_2 with weight at most k traverses $r > 1$ edges. Assume that the first edge on the path is $e = (v_1, v_3)$, and $w(e) = w$. Note that a minimal length path from v_3 to v_2 with weight at most $k-w$ is identical to the minimal length path from v_1 to v_2 with weight at most k, with the edge e removed. For if there were a path of smaller length, it would produce a path of smaller length from v_1 to v_2 with weight at most k, by attaching e.

Since the minimal length path with weight at most $k-w$ from v_3 to v_2 traverses $r-1$ edges, $L(v_3, v_2, k-w)$ contains the proper value after $r-1$ iterations (induction). By the rth iteration, v_3 sends a message to v_1 "$L(v_1, v_2, k) \leq L(v_3, v_2, k-w) + \ell(e)$". Using this information, v_1 updates $L(v_1, v_2, k)$ in the rth interation and the lemma (and theorem) is proved. ■

Running Time of the Algorithm

The number of machine steps is at most $O(n^2)$ times the running time per iteration per node (Theorem 1). In step 1, of the algorithm each node sends out at most n messages (at most n neighbors) for each entry in its tables of size n^2b. Since each message is of size $O(\log nb)$, step 1 takes at most time $O(n^3b \log nb)$. Similarly, each node receives at most $O(n^3b)$ messages. Unfortunately in step 2c, each node may do $O(nb)$ work per message for a total of $O(n^4b^2)$ machine steps per iteration.

The amount of time for 2c is reduced as follows. If a message "$T(v_1, v_2, k) \leq x$" arrives, only $T(v_1, v_2, k)$ is updated but not $T(v_1, v_2, k')$ for $k' > k$. After all $O(n^3b)$ messages have been handled, v_1 makes one pass through its tables to insure $T(v_1, v_2, k') \leq T(v_1, v_2, k)$ if $k' \geq k$. Thus $O(n^2b)$ additional time is required, but each iteration requires only $O(n^3b \log nb)$.

The tables actually solve more than MCP. For a given W, one may find the *minimal length* path with weight at most W. One may also find the *minimal weight* path with length at most L. This is accomplished by searching $L(v_1, v_2, *)$ for the smallest value of k with the property $L(v_1, v_2, k) \leq L$. One may also search the table to minimize linear combinations of the form $cL(p) + dW(p)$. If p minimizes $cL(p) + dW(p)$, then p is also the minimum length path with weight $W(p)$.

In computer network situations, it is not enough to know whether some path satisfies the constraints. The path itself must be explicitly determined so that communication between nodes may take place on this path. This information does not exist in the tables as defined, but may be conveniently introduced as follows.

When v_1 discovers that $L(v_1, v_2, k)$ needs to be updated due to a short path (in terms of length) that goes to v_2 via v_3, v_1 associates with $L(v_1, v_2, k)$: "NEXT-NODE $(v_1, v_2, k) = v_3$". The interpretation is that the minimal length path from v_1 to v_2 with weight at most k has as its first edge (v_1, v_3). Using this, it is easy to determine the entire path from v_1 to v_2. The path with weight at most k, and length $L(v_1, v_2, k)$ has as its first node NEXT-NODE $(v_1, v_2, k) = v_3$. The second node is NEXT-NODE $(v_3, v_2, k-w(v_1, v_3)) = v_4$. Third is NEXT-NODE $(v_4, v_2, k-w(v_1, v_3) - w(v_3, v_4))$ and so on.

3. APPROXIMATING THE MCP PROBLEM

The approach of the last section may at times be inappropriate. If b is large it is quite expensive in time and space. Even for small b, n^5 is quite large for moderate values of n. An algorithm that approximates the constraints, and runs quickly is thus appropriate.

To quickly find paths in a network, a natural algorithm is to minimize $\alpha L(p) + \beta W(p)$ where $\alpha, \beta \in \mathbb{Z}^+$. To solve this efficiently one merely defines a length function $\ell': E \rightarrow \mathbb{Z}^+$ by $\ell'(e) = \alpha\ell(e) + \beta w(e)$. The path that minimizes $L'(p)$ also minimizes $\alpha L(p) + \beta W(p)$, since $L'(p) = \alpha L(p) + \beta W(p)$ for every p.

The following objective function, f, measures the amount by which a particular path violates the constraints. It has the additional property that if any feasible solutions to an instance of MCP exist, then all feasible solutions minimize f:

$$f(p) = \max(L(p), L) + \max(W(p), W) \tag{1}$$

The best possible value of f is L+W, which is achieved only at feasible solutions. A path p, is not considered any better if L(p) is much smaller than L, rather than slightly smaller. If a constraint is violated, the path's value depends linearly on the "extent of constraint violation." Any algorithm that minimizes f(p) solves MCP. There is a feasible solution iff some path p has f(p) = L+W.

4. A LIMITED INFORMATION ALGORITHM

As a means of introducing our techniques we give a simple way to approximate f. This way does not use the values of L and W in the problem instance, but finds the same path for all L and W. A possible application is to provide a single path between two nodes in a network, which performs reasonably well for different sets of constraints. The algorithm is to minimize L(p) + W(p). Assume W ≥ L.

Theorem 2. Fix a graph G = (V, E), length function, ℓ, weight function, w, and bounds W, L as above. Let $v_1, v_2 \in V$, and let p* be the minimum-f-path between v_1 and v_2. Let p′ be the path from v_1 to v_2 that minimizes L(p) + W(p). Then $\frac{f(p')}{f(p^*)} \leq 1 + \frac{W}{L+W} \leq 2$.

Proof. Define g(p) = L(p) + W(p). Clearly f(p) ≥ g(p) for all p. Assume p is not feasible, i.e. L(p) > L or W(p) > W. Then g(p) + max(L,W) ≥ f(p). This follows from g(p) = L(p) + W(p), and the fact that f(p) is either L(p) + W, W(p) + L or W(p) + L(p).

If p′ is feasible then $\frac{f(p')}{f(p^*)} = 1$ and the theorem is proved. Assume p′ is not feasible. Then:

$$\frac{f(p')}{f(p^*)} \leq \frac{g(p') + \max(L,W)}{f(p^*)} \leq \left(\frac{g(p')}{g(p^*)} + \frac{W}{L+W}\right) \leq \left(1 + \frac{W}{L+W}\right) \leq 2 \tag{2}$$

The first inequality follows from f(p′) ≤ g(p′) + max(L,W). The second follows from f(p*) ≥ g(p*), f(p*) ≥ L+W, and max(L,W) = W. The third inequality follows from g(p′) ≤ g(p*), since p′ minimizes g. ■

Note that the bound of $1 + \frac{W}{L+W}$ is tight, in that for every pair (L, W), the algorithm does at times perform that poorly. To see this, consider the graph of Figure 1 with lengths and widths as labelled. The path that minimizes L(p) + W(p) is p = (A, B, C) with g(p) = L+W−ε and f(p) = L+2W−ε. However f(A, C) = L+W. Thus

$$\frac{f(p)}{f(A,C)} = \frac{L+2W-\varepsilon}{L+W} = 1 + \frac{W-\varepsilon}{L+W} \tag{3}$$

As $\varepsilon \to 0$, the ratio approaches $1 + \frac{W}{L+W}$.

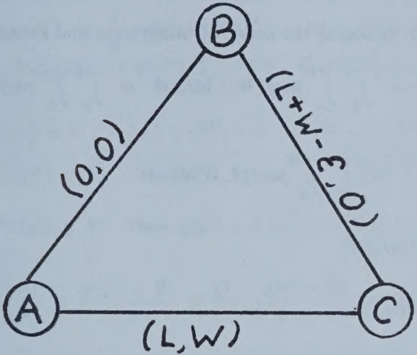


Figure 1

In evaluating a heuristic, it is useful to know its average behavior. For example, if a single path is used for many values of L and W, the average performance is of interest. Unfortunately, the average depends on "the average graph to be found in practice" or, "average length and weight functions". Generally, such analyses must be made with knowledge of the particular application, e.g., if a graph represents a computer network, then one might assume that the average degree of a node is small. The following describes one general definition of average behavior.

Assume that L and W are random variables with joint probability density function h(L, W). Assume that the worst case ratio of an algorithm, A, is perf(L, W)(in terms of minimizing f). Then the *average worst case performance of A* is defined by:

$$\overline{perf} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(L,W)\, perf(L,W)\, dL\, dW \qquad (4)$$

Since we do not know how L and W will be distributed in general, we use simple distributions just to get some feeling for average behavior. To do this, we first redefine the MCP problem. Assume that all lengths and weights are arbitrary real numbers, and that W and L are independent and uniformly distributed in [0, 1]. Thus (4) may be rewritten as:

$$\overline{perf} = \int_0^1 \int_0^1 perf(L,W)\, dL\, dW \qquad (5)$$

It is our contention that the above provides some reasonable measure of "average worst case performance". In the sequel we use a different version of (5). Recall that perf(L,W) =

perf(W,L) for the algorithm discussed above. It turns out that this is true for all algorithms in this paper (since it is essentially a renaming of L and W). Thus

$$\int_0^1 \int_0^1 perf(L,W)dLdW = \int_0^1 \int_0^W perf(L,W)dLdW + \int_0^1 \int_W^1 perf(L,W)dLdW \qquad (6)$$

Using perf (L,W) = perf (W,L), changing the order of integration and renaming the variables;

$$\int_0^1 \int_W^1 perf(L,W)dLdW = \int_0^1 \int_w^1 perf(W,L)dLdW = \int_0^1 \int_0^L perf(W,L)dWdL \qquad (7)$$

$$= \int_0^1 \int_0^W perf(L,W)dLdW$$

Thus $\overline{perf} = 2\int_0^1 \int_0^W perf(L,W)dLdW.$

(In the sequel ℓn refers to $\log_e$.)

Theorem 3. Let $\overline{perf}$ be as above (average performance of minimizing L(p) + W(p) where L and W are uniform in [0,1]). Then $\overline{perf} = 1 + \ell n2 \approx 1.6931$.

Proof. In suffices to calculate

$$\int_0^1 \int_0^W \left(1 + \frac{W}{L+W}\right) dL = \int_0^1 L + W\ell n(L + W)\Big]_0^W dW \qquad (8)$$

$$= \int_0^1 W + W\ell n2 = \frac{1}{2}(1 + \ell n2).$$

The value of $\overline{perf}$ is twice the above quantity. ■

Having introduced the basic technique of approximating f by doing a shortest path calculation, and the basic method of evaluating average behavior, we turn to improved algorithms. *The algorithms will use L and W to decide which shortest path calculation to make.*

5. AN IMPROVED ALGORITHM

The approach used here is to minimize L(p) + dW(p). The value d is chosen on the basis of L and W. An algorithm is described for choosing d, if some feasible solution exists. (If a feasible solution *does* exist, one might be especially concerned about finding a reasonable approximation.) A different algorithm that has better overall behavior if no feasible solution exists but is not as good when feasible solutions exist is given in [5]. The worst case and average worst case behavior is given.

Lemma 2. Let G, ℓ, w, L, W, v_1, v_2 and p* as in Theorem 2. Let p′ be the path from v_1 to v_2 that minimizes L(p) + dW(p). If p* is feasible, then $\frac{f(p')}{f(p^*)} \le 1 + \max\left(\frac{Wd}{W+L}, \frac{\frac{L}{d}}{W+L}\right)$.

Proof. The proof is divided into four cases:

Case 1. $L(p') \le L$, $W(p') \le W$. In this case $f(p') = f(p^*)$

Case 2. $L(p') > L$, $W(p') > W$. This case cannot happen, since it implies $L(p') + dW(p') > L(p^*) + dW(p^*)$.

Case 3. $L(p') \le L$, $W(p') > W$. Define $g(p) = L(p) + dW(p)$. Then $g(p') = L(p') + dW(p')$ and $\frac{1}{d} \cdot g(p') = \frac{1}{d} L(p') + W(p')$. Also $f(p') = L + W(p')$. Thus $f(p') \le \frac{1}{d} g(p') + L$. Also $g(p') \le g(p^*) \le L+dW$. Combining these inequalities yields:

$$\frac{f(p')}{f(p^*)} \le \frac{\frac{1}{d} g(p') + L}{f(p^*)} \le \frac{\frac{1}{d}(L + dW) + L}{f(p^*)} = \frac{W + L(\frac{1}{d} + 1)}{W + L} = 1 + \frac{\frac{L}{d}}{L + W} \tag{9}$$

Case 4. $L(p') > L$, $W(p') \le W$. Then $g(p') = L(p') + d\,W(p')$, $f(p') = L(p') + W$ and $f(p') \le g(p') + W$. Thus

$$\frac{f(p')}{f(p^*)} \le \frac{g(p') + W}{f(p^*)} \le \frac{(L + dW) + W}{W + L} = 1 + \frac{dW}{L + W} \tag{10}$$

From (9) and (10), Lemma 2 follows immediately. ■

For a given pair (L,W), the value of d that provides the best performance is the one that minimizes $\max\left(1 + \frac{dW}{L+W}, 1 + \frac{\frac{L}{d}}{L+W}\right)$. Since the first term is an increasing function of d and the second a decreasing function of d, the max is minimized when they are equal, i.e. when $dW = \frac{L}{d}$ or $d = \sqrt{L/W}$. Thus the algorithm of minimizing $L(p) + dW(p)$ gets best performance when using $d = \sqrt{L/W}$. In that case:

Theorem 4. Let G, ℓ, w, L, W, v_1, v_2 and p^* as in Theorem 2, and p' defined based on $d = \sqrt{L/W}$. Then $\frac{f(p')}{f(p)} \le 1 + \frac{\sqrt{WL}}{W+L} \le 1.5$.

Proof. Substitute $d = \sqrt{L/W}$ in $1 + \max\left(\frac{Wd}{W+L}, \frac{\frac{L}{d}}{W+L}\right)$ for the first inequality. To prove that $\frac{1}{2} \ge \frac{\sqrt{WL}}{W+L}$, note $(\sqrt{W} - \sqrt{L})^2 \ge 0$, thus $W - 2\sqrt{W}\sqrt{L} + L \ge 0$, thus $W + L \ge 2\sqrt{WL}$, thus $\frac{1}{2} \ge \sqrt{WL}/(W + L)$. ■

Theorem 5. With notation as above, the average worst case behavior of minimizing $L(p) + (\sqrt{L/W})W(p)$ when some feasible solution exists is $3 - \frac{\pi}{2} = 1.4292$.

Proof. It suffices to calculate

$$\int_0^1 \int_0^W \left(1 + \frac{\sqrt{WL}}{W+L}\right) dL dW = \frac{1}{2} + \int_0^1 \int_0^W \frac{\sqrt{WL}}{W+L}\, dL dW \tag{11}$$

Let $x = \sqrt{L}, x^2 = L, 2xdx = dL$. Then (11) becomes

$$\frac{1}{2} + \int_0^1 \int_0^{\sqrt{W}} \frac{\sqrt{W}\, 2x^2\, dx}{W + x^2}\, dx dW = \frac{1}{2} + \int_0^1 2\sqrt{W}\Big(x - \sqrt{W}\ \tan^{-1} \frac{x}{\sqrt{W}}\Big]_0^{\sqrt{W}}\Big) dW$$

$$= \frac{1}{2} + \int_0^1 2W\Big(1 - \frac{\pi}{4}\Big) dW = \frac{3}{2} - \frac{\pi}{4}. \qquad (12)$$

The integration with respect to x can be done using a table of integrals [6]. As above, $\overline{perf}$ is just twice (12). ■

6. A GENERAL ALGORITHM

In [5] a different algorithm to approximate MCP is given. It is at most $\frac{\sqrt{5}+1}{2} \approx 1.618$ times worse than optimal. Its average worst case behavior is 1.5103 times worse than optimal. The outline of the algorithm is identical to that of Section 5, one minimizes L(p) + dW(p), but the parameter d is chosen to insure good behavior even if no feasible solution exists. The details are omitted here.

7. SUMMARY

We have taken an NP-complete problem with important practical applications and shown how to solve it for all practical purposes. An algorithm solves it in polynomial time if the lengths and weights have a small range of values. We have indicated how to approximate the problem if one only had a small amount of time available.

ACKNOWLEDGEMENTS

The author acknowledges useful discussions with F. H. Moss.

REFERENCES

1. W. D. Tajibnapis, "A Correctness Proof of a Topology Information Maintenance Protocol for a Distributed Computer Network", *CACM* 20, 7, July 1977, 477-85.
2. P. M. Merlin and A. Segall, "A Failsafe Distributed Routing Protocol", IEEE Transactions on Communications, *COM-27,* Sept. 1979, pp. 1230-37.
3. M. R. Garey and D. S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-Completeness,* W. H. Freeman, San Francisco, (1979).
4. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms,* Addison-Wesley, Reading, 1974.
5. J. M. Jaffe, "Algorithms for Finding Paths with Multiple Constraints", *IBM Research Report*, RC8205, April 1980.
6. H. B. Dwight, *Tables of Integrals and other Mathematical Data.*

ON FINDING SEVERAL SHORTEST PATHS IN CERTAIN GRAPHS*

(Preliminary Report)

ZVI M. KEDEM
Department of Computer Science
S.U.N.Y. at Stony Brook
Stony Brook, New York 11794

and

HENRY FUCHS
Department of Computer Science
University of North Carolina at Chapel Hill
Chapel Hill, North Carolina 27514

ABSTRACT

Given a graph, the problem of finding shortest paths from $u_1^{(i)}$ to $u_2^{(i)}$ for a set of pairs of vertices $\{(u_1^{(i)}, u_2^{(i)}) \mid i=1,\ldots,m\}$ is considered.

A fast algorithm for a special class of problems is presented. The algorithm is applied to the Circular String-to-String Correction Problem.

---

*Research partially supported by NSF under Grants Number MCS-7901168 and MCS-7902593, and facilitated by the use of Theory Net, NSF Grant Number MCS-7801689.

GENERAL THEOREM

The problem of finding shortest paths in graphs has been studied extensively. Two main variants have been of interest:

- I. For one pair of vertices (u,w) in the graph find a shortest path from u to w.
- II. For all pairs of vertices (u,w) in the graph find a shortest path from u to w.

Well-known methods exist for dealing with these variants.

We consider in this paper another possible variant, which falls between the two variants above:

- III. For several pairs of vertices $\{(u_1^{(i)}, u_2^{(i)}) \mid i=1,\ldots,m\}$ in the graph find a shortest path from each $u_1^{(i)}$ to its corresponding $u_2^{(i)}$.

Let then $G = \langle V,E,L\rangle$ be an undirected or a directed and acyclic, labeled graph, where V is the set of vertices, E the set of edges, and L the labeling function assigning to each element of E a real number (length), which will be non-negative if G is undirected.

Variant I is normally solved by algorithms which require $O(|V|^2)$ or $O(|E| \cdot \log_2 |V|)$ operations in the worst case, and variant II is normally solved by algorithms which require $O(|V|^3)$ or $O(|E| \cdot \log_2 |V|)$ operations in the worst case. (See e.g. [AHU].)

A solution of variant III may be obtained by either solving an instance of variant I for each of the given pairs or by solving variant II and extracting the required shortest paths. In this paper we will describe a method which can be used to design efficient algorithms for solving variant III. We restrict the development here only to directed acyclic graphs, although the results also hold for undirected graphs. As we will be concerned here solely with directed graphs we will refer to edges as arcs.

For a graph $G = \langle V,E\rangle$ and $V' \subseteq V$ we shall denote by $G(V')$ the subgraph of G spanned by V'. A path in G will be written as a sequence of vertices; a path will also be considered as a subgraph of G spanned by the traversed arcs.

Assume that shortest paths from $u_1^{(i)}$ to $u_2^{(i)}$ for $i = 1,2$ are to be found. If a shortest path P(1) from $u_1^{(1)}$ to $u_2^{(1)}$ is found first then, as we shall see, the search for a shortest path P(2) from $u_1^{(2)}$ to $u_2^{(2)}$ may be limited to a subgraph of G. If one has to find more than two shortest paths, then possibly even "smaller" subgraphs of G may be considered during the search for some paths.

Theorem 1: Let $P_0 = (u_1, u_2, \ldots, u_p)$ be a shortest path from u_1 to u_p in a directed acylic graph G. Let $V_0 \hat{=} \{u_1, u_2, \ldots, u_p\}$, and let the weak (undirected) components of $G(V-V_0)$ be spanned by the (pairwise disjoint) sets of vertices $V_1, V_2, \ldots, V_r$. If $v_a \in V_i$ and $v_b \in V_j$ for $0 \leq i, j \leq r$ are such that v_b is reachable from v_a, then there exists a shortest path $P = (v_a = w_1, w_2, \ldots, w_q = v_b)$ in G from v_a to v_b such that

1. $w_k \in V_0 \cup V_i \cup V_j$, $k = 1, 2, \ldots, q$,
2. if $w_k, w_{k+1} \in V_0$ for some $k = 1, 2, \ldots, q-1$, then $w_k = u_\ell$ and $w_{k+1} = u_{\ell+1}$ for some $\ell = 1, 2, \ldots, p-1$. ■

For P_0, V_0, v_a, V_i, v_b, V_j as in Theorem 1, $G(P_0, v_a, v_b)$ will denote the subgraph of G to which the search for a shortest path from v_a to v_b and be limited. More formally, if $\tilde{V} = V_0 \cup V_i \cup V_j$ and $P_0 = \langle V_0, E_0 \rangle$ (E_0 consists of the arcs traversed by the path P_0) then $G(P_0, v_a, v_b) \hat{=} \langle \tilde{V}, (\tilde{V}^2 \cap E) - (V_0^2 - E_0) \rangle$.

Theorem 1 can be used to design efficient algorithms for finding shortest paths for several pairs of vertices. Indeed, assume that are to find shortest paths from u_1 to u_p and from v_a to v_b. Once a shortest path P_0 from u_1 to u_p has been found, one may search for a shortest path from v_a to v_b in $G(P_0, v_a, v_b)$ only. If additional pairs of vertices are given, then possibly even smaller subgraphs of G have to be considered. The subgraphs to which the search may be limited, $G(P_0, v_a, v_b)$ and the appropriate subgraphs for the other pairs, may easily be computed in a linear number of operations but any possible savings greatly depend on G and the given pairs of vertices, as in the general case there is no reason to assume that these subgraphs are "smaller" than the original graph G.

APPLICATION TO PLANAR GRAPHS

We shall now develop the use of the Theorem for certain cases in which it can be assured that the size of the subgraphs to be considered rapidly decreases as paths for additional pairs of vertices are found. As there can be no path in G whose terminals lie in distinct components of a graph, we shall assume, without loss of generality, that G consists of a single component.

Let G now be a planar graph, which is identified with an actual embedding in the plane. A sequence of vertices $(u_1, u_2, \ldots, u_r)$ will be called a circuit in G if and only if it is an (undirected) walk in G and $u_r = u_1$. Each such circuit $\alpha = (u_1, u_2, \ldots, u_r)$ defines a closed curve ℓ_α on the plane which is defined by the sequence of the edges "traversed" by α. Let $V^{(\alpha)}$ be the set of those vertices of G which lie on or inside ℓ_α. We define G_α, a subgraph of G, to be the subgraph of $G(V^{(\alpha)})$ obtained from $G(V^{(\alpha)})$ by removing all those edges which lie outside of ℓ_α. (See Fig. 1.) A circuit α will be called non-crossing if and only if the corresponding curve ℓ_α does not cross itself. (It may "touch" itself.)

Without loss of generality, we may assume that an arbitrary contour is the contour of the infinite region of G and thus the following development, even though formulated for the contour Γ_0 of the infinite region, holds with appropriate changes for the contour of an arbitrary region.

A sequence $(z_1,z_2,\ldots,z_h)$ of vertices in G is a list for Γ_0 if and only if it is a non-crossing circuit in G and h is minimal such that the set $\{z_i | i=1,\ldots,h\}$ is the set of all the vertices in Γ_0.

Let now $(z_1,z_2,\ldots,z_h)$ be a list for Γ_0 and let $P = (z_s=w_1,w_2,\ldots,w_r = z_t)$ be a path from z_s to z_t for $1\leq s,t\leq h$. Note that as the graph is acyclic P cannot "cross" itself. Define the sequences

$$\Gamma_0 \oplus P \mathrel{\hat{=}} \begin{cases} (z_s,z_{s+1},\ldots,z_t=w_r,w_{r-1},\ldots,w_1=z_s) & ; \ s \leq t \\ (z_s=w_1,w_2,\ldots,w_r=z_t,z_{t+1},\ldots,z_s) & ; \ s > t \end{cases}$$

$$P \oplus \Gamma_0 \mathrel{\hat{=}} \begin{cases} (z_s=w_1,w_2,\ldots,w_r=z_{t+1},\ldots,z_h=z_1,\ldots,z_s) & ; \ s \leq t \\ (z_s,z_{s+1},\ldots,z_h=z_1,\ldots,z_t=w_r,w_{r-1},\ldots,w_1=z_s); & s > t \end{cases}$$

$\Gamma_0 \oplus P$ and $P \oplus \Gamma_0$ are non-crossing circuits and are furthermore lists for the contours of the infinite regions of the subgraphs $G_{\Gamma_0 P}$ and $G_{P\,\Gamma_0}$ of G respectively. (See Fig. 2.)

Corollary 1: Let $(z_1,z_2,\ldots,z_h)$ be a list for Γ_0 and let z_s and z_t for $1\leq s\leq t\leq h$ be terminals of a shortest path P. If a and b are any integers such that $\{a,b\} \subseteq \{s,s+1,\ldots,t\}$ (or $\{a,b\} \subseteq \{t,t+1,\ldots,h,1,\ldots,s\}$ respectively), and z_b is reachable from z_a, then there exists a shortest path (in G) from z_a to z_b which is wholly contained in $G_{\Gamma_0 \oplus P}$ (or $G_{P\oplus\Gamma_0}$ respectively). ■

The Corollary defines in a natural way a family of problems for which it can be utilized for the design of a "divide and conquer" algorithm.

Let again $(z_1,z_2,\ldots,z_h)$ be a list for Γ_0. For some $m > 0$ let ϕ be a nondecreasing function from $\{1,2,\ldots,2m\}$ to $\{1,2,\ldots,h\}$ and let δ be a function from $\{1,2,\ldots,m\}$ to $\{0,1\}$. For every k, $k=1,2,\ldots,m$, let $z_k^{(2)} = z_{\phi(2m+1-k)+\delta(k)\cdot(\phi(k)-\phi(2m+1-k))}$ be reachable from $z_k^{(1)} = z_{\phi(k)+\delta(k)\cdot(\phi(2m+1-k)-\phi(k))}$. (See Fig. 3) We are to find $P(1),P(2),\ldots,P(m)$ where $P(k)$ is a shortest path from $z_k^{(1)}$ to $z_k^{(2)}$. ($P(k)$ will be a shortest path whose terminals are $z_{\phi(k)}$ and $z_{\phi(2m+1-k)}$.)

We shall now present an algorithm for finding $P(1),P(2),\ldots,P(m)$. We assume that a procedure SINGLEPATH(k,Γ) which finds $P(k)$ in the subgraph G_Γ of G has already been defined. A single invocation of the algorithm SEVERALPATHS(i,j,Γ) finds $P(i),P(i+1),\ldots,P(j)$:

```
Algorithm SEVERALPATHS(i,j,Γ)
1.  k    :=  ⌊(i+j)/2⌋;
2.  P(k)  := SINGLEPATH(k,Γ);
3.  if i< k then SEVERALPATHS (i,k-1,P(k)⊕Γ);
4.  if k < j then SEVERALPATHS(k+1,j,Γ⊕P(k));
```

All m shortest paths $P(1), P(2),\ldots, P(m)$ are found as a result of a single invocation of SEVERALPATHS$(1,m,\Gamma_0)$. In this invocation, line 2 finds a shortest path $P(h)$ in $G_{\Gamma_0}=G$, where $h = \lfloor(1+m)/2\rfloor$. This lets the algorithm apply the "divide and conquer" approach to finding the remaining paths. Indeed, line 3 finds $P(1),P(2),\ldots,P(h-1)$ in $G_{P(h)\oplus\Gamma_0}$ and line 4 finds $P(h+1),P(h+2),\ldots, P(m)$ in $G_{\Gamma_0\oplus P(h)}$. The Corollary assures that the algo- will terminate with correct results.

The procedure SINGLEPATH can be easily implemented by using say, Dijkstra's algorithm in G, and abandoning the search along any path which crosses Γ. If the data structure defining the graph is implemented suitably by associating with each vertex a list of its neighbors in a counter-clockwise order, such "crossings" can be easily detected.

Let us now consider the time required by SEVERALPATHS (i,j,Γ) (Γ will be $P(j+1)\oplus\Gamma_0\oplus P(i-1)$ where $P(0)$ and $P(m+1)$ are null. For a connected subgraph H of G we shall denote by $T(H)$ the number of operations required to find any single shortest path in H using SINGLEPATH. If $|H|$ is the number of arcs in H then we can write $T(H)\simeq t(|H|)$ for an appropriate function t. (For Dijkstra's algorithm $t(x) = O(x\cdot\log_2 x)$ as in a connected planar graph the number of vertices is bounded from below and from above by linear functions of $|H|$). We know that $t(0) = 0$ and $t(x)$ grows at least linearly with x and thus we shall assume that if $0 \leq x_1,x_2$ then $t(x_1) + t(x_2) \leq t(x_1+x_2)$. Furthermore, if t grows faster than linearly, then $t(x_1) + t(x_2) < t(x_1+x_2)$, for $0 < x_1,x_2$.

For ease of notation, assume that $m=2^\mu -1$ for some μ. One can show that the number of operations, OPS, is bounded by the formula:

$$\mathrm{OPS} \leq \sum_{s=1}^{\mu} t\ (|G| + \sum_{j=1}^{2^{s-1}-1} |\ P(j\cdot 2^{\mu-s+1}) \leq$$

$$t\ (\log_2\ (m+1)\cdot |G| + \sum_{s=1}^{\log_2(m+1)} \sum_{j=1}^{2^{s-1}-1} |\ P(2j(m+1)/2^s)\ |).$$

If $| P(2j(m+1) / 2^s |$'s are not known (or at least no upper bound is known) we conjecture that

$$\text{OPS} \leq \sum_{s=1}^{\mu} 2 \cdot t(|G|) = 2 \log_2 (m+1) \cdot t(|G|)$$

CIRCULAR STRING-TO-STRING CORRECTION PROBLEM

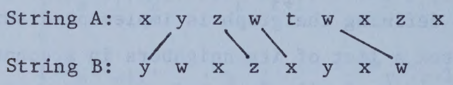
In [WF] a fast algorithm to solve the String-to-String Correction Problem has been presented. Following a suggestion by M. Fischer we apply our algorithm to solve the correction problem for "circular" strings. We assume in the sequel that the reader is familiar with [WF]. Briefly, the paper shows how to utilize dynamic programming to find a minimum cost edit sequence taking a string $A=A_1A_2\ldots A_I$ to a string $B=B_1B_2\ldots B_J$. This problem can also be solved, with the same complexity, by finding a shortest path in an appropriate planar graph.

Indeed consider the graph $G= \langle V,E,L \rangle$ where

$$V = \{v_{ij} \mid 0 \leq i \leq I,\ 0 \leq j \leq J\},$$

$$\langle v_{ij}, v_{k\ell} \rangle \in E \Longleftrightarrow (i,j) \neq (k,\ell),\ 0 \leq k-1 \leq i \leq k \leq I,\ 0 \leq \ell-1 \leq j \leq \ell \leq J.$$

Then every possible edit sequence taking A to B corresponds to a path in G from v_{00} to v_{IJ}. For example, the diagram:



copied from [WF] corresponds to the path drawn in Fig. 4.

We now associate the following length function with the arcs of G, (γ is defined in [WF]).

$$L(\langle v_{ij}, v_{k\ell} \rangle) = \begin{cases} \gamma(A_k \to B_\ell) & ; \quad k=i+1,\ \ell=j+1 \\ \gamma(A_k \to \Lambda) & ; \quad k=i+1,\ \ell=j \\ \gamma(\Lambda \to B_\ell) & ; \quad k=i\ \ ,\ \ell=j+1 \end{cases}$$

Then a minimum cost edit sequence corresponds to a shortest path.

Extend now the problem to consider circular strings. Namely, assume that one is permitted at no cost to circularly shift each string by an arbitrary number of positions before commencing the edit sequence. Formally, we wish to minimize the edit cost for taking

$$A_iA_{i+1}\ldots A_IA_1\ldots A_{i-1} \text{ to } B_jB_{j+1}\ldots B_JB_1\ldots B_{j-1}$$

over all possible i and j. This problem is transformed into finding min {I,J} shortest paths in an appropriate planar graph, similar to the one used in [FKU] to solve the problem of optimal surface reconstruction.

Define $G^1=(V^1,E^1,L^1)$ where

$V^1=\{v_{ij} \mid 0\leq i\leq 2I-1, 0\leq j\leq J\}$,

$<v_{ij},v_{k\ell}> \varepsilon E^1 \iff (i,j)\neq(k,\ell),\ 0\leq k-1\leq i\leq k\leq 2I-1, 0\leq \ell-1\leq j\leq \ell\leq J$, and

$L^1<v_{ij},v_{k\ell}>) = L(<v_{i(\mathrm{mod}\ I),j},v_{k(\mathrm{mod} I),\ell}>)$. (See Fig. 5.)

Let P_i for $i=0,1,\ldots,I-1$ denote a shortest path in G^1 from v_i to $v_{i+I,J}$. Then a shortest path in $\{P_0,P_1,\ldots,P_{I-1}\}$ corresponds to a minimum cost edit sequence for the circular strings A and B. Using the algorithm SEVERALPATHS, it can be computed in the number of operations less than $\log_2(I+1) + 2$ times that required to solve the standard String-to-String Correction Problem. (We assumed that $I\leq J$. If $J<I$ a simple modification is required.)

REFERENCES

[AHU] Aho, A.V., J.E. Hopcroft, and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley (1974).

[FKU] Fuchs, H., Z.M. Kedem, and S.P. Uselton, Optimal Surface Reconstruction from Planar Contours, CACM 10 (1977), 693-701.

[WF] Wagner, R.A. and M.J. Fischer, The String-to-String Correction Problem, JACM 21 (1974), 168-173.

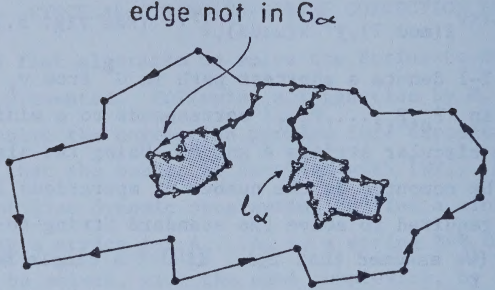

edge not in Gα
ℓα

Figure 1

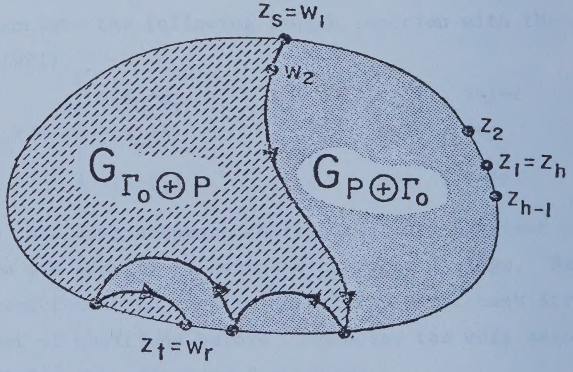

Zs=W1
W2
Z2
Z1=Zh
Zh-1
GΓo⊕P
GP⊕Γo
Zt=Wr

Figure 2

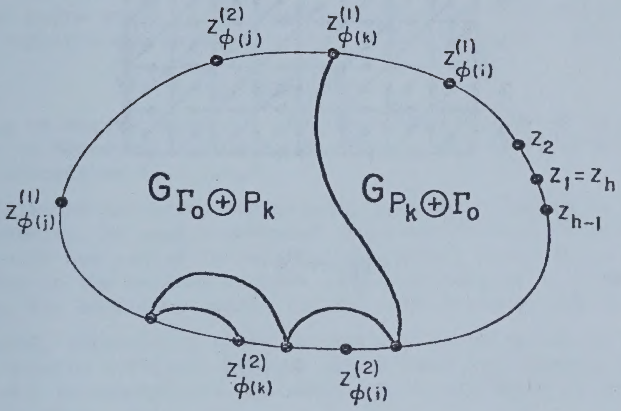

$z^{(2)}_{\phi(j)}$
$z^{(1)}_{\phi(k)}$
$z^{(1)}_{\phi(i)}$
z_2
$z_1 = z_h$
z_{h-1}
$z^{(1)}_{\phi(j)}$
$G_{\Gamma_0 \oplus P_k}$
$G_{P_k \oplus \Gamma_0}$
$z^{(2)}_{\phi(k)}$
$z^{(2)}_{\phi(i)}$

Figure 3

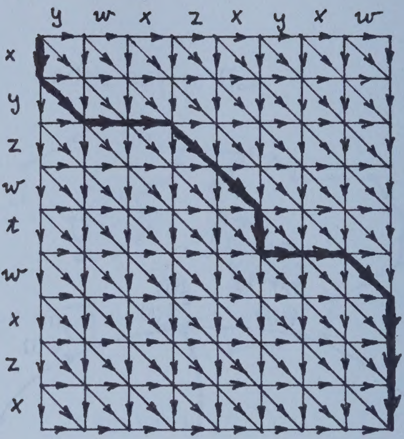

y w x z x y x w
x
y
z
w
t
w
x
z
x

Figure 4

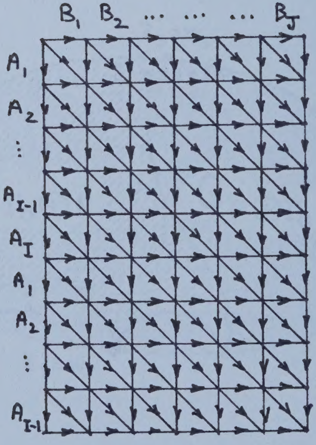

B₁ B₂ … … … B_J
A₁
A₂
⋮
A_{I-1}
A_I
A₁
A₂
⋮
A_{I-1}

Figure 5

EFFICIENT DISTRIBUTED ALGORITHMS FOR FINDING CENTERS AND MEDIANS IN NETWORKS (Preliminary version)

E. KORACH*, D. ROTEM**, N. SANTORO***

ABSTRACT

In this paper, we consider the problem of determining in a distributed fashion the centers and the medians of a network. Lower bounds on the time, needed to solve these problems are proved. Algorithms that achieve those bounds for tree networks are presented; the number of exchanged messages is linear in the number of nodes. We extend these techniques to work on general networks in $O(n)$ time units exchanging $O(n \cdot e)$ messages, where n is the number of nodes and e is the number of edges in the network. In addition, a comparison with a simple heuristic approach is included.

1. INTRODUCTION

Recently we are witnessing a wide and growing interest in the design and analysis of decentralized networks (for example MERIT or ARPANET) and distributed algorithms [1,2,5,6,8].

In distributed networks, topological information plays an important role. For example, in packet-switched store-and-forward networks, packets leaving a source are routed to intermediate nodes; thus, it is essential for every node in the network to have some knowledge of the topology of the network (e.g. the adjacency matrix [14,15], the distance matrix [5], etc.)

In general, information about the network topology can be usefully employed to develop efficient network algorithms, for example, the number of steps needed to synchronize the nodes can be minimized if the center of the network is known [10].

Unfortunately, topology information cannot be taken into account once and for all at design time. In fact, several unpredictable factors make the topology vary in time, e.g. power failure, computer crash, link reactivation, etc. Therefore this information must be redetermined whenever it is needed.

In this paper we present and analyze distributed algorithms for locating centers and medians of a network. Such location algorithms are important in the design of routing mechanisms which provide broadcast with small delay [16].

The organization of the paper is as follows. In Section 2 the model is described. In Section 3, we consider the special important case of tree networks. A basic algorithm is presented: this algorithm activates all nodes in the network. An activated node in turn reports some information which is eventually collected at a certain node. It is shown that by processing the collected information we can design efficient algorithms to determine the centers and the medians of a tree.

---

* Department of Combinatories and Optimization, University of Waterloo, Waterloo, Ontario, Canada, N2I 3Gl.

** Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3Gl

*** Department of Computer Science, University of Ottawa, Ottawa, Ontario, Canada, KlN 9B4

Furthermore we find a lower bound on the time required to find a center (or a median) in a tree and prove that our algorithms achieve this lower bound and are therefore time optimal. The number of messages exchanged in both algorithms is shown to be linear in the number of nodes of the tree.

In Section 4 we extend the results of Section 3 to general networks. The idea here is to construct shortest path spanning trees for each node, and then locate the center or the median of the network by conducting a competition among the nodes of the network.

The exact solutions of our algorithms are compared with the approximate solutions obtained by heuristics that instead choose the center or the median of an arbitrary spanning tree. Achievable bounds on the accuracy of such approximate solutions show that they can be far from optimal. The proposed algorithms for finding the center and the median require at most $5.r(G)$ and $5.d(G)$ units of time respectively where $r(G)$ is the radius and $d(G)$ is the diameter of a network G. We also give upper bounds on the number of exchanged messages. The prooves of all theorems, lemmas and properties can be found in [17].

2. THE MODEL

In this section we describe the framework and define some terms that are used throughout the paper.

A communication network can be represented as a graph $G = (V(G),E(G))$ where $V(G)$ is a set of nodes and $E(G) \subseteq V(G)\times V(G)$ is a set of arcs. An arc $(i,j)\in E(G)$ represents a bidirectional communication link between nodes i and j.

We denote by $INF(i,t)$ the information about the network stored at node i at time t. We assume that for $i\neq j$ $INF(i,0) \not\subseteq INF(j,0)$. i.e., before any exchange of information occurs, no node i knows the information of node j, for all $i,j\in V(G)$. Each node i makes a decision $D(i,t)$ at time t which is a function of the total information known to i at this time, i.e. $D(i,t) = f(totalINF(i,t))$. For each node i, $INF(i,0)$ includes a list of its neighbours $N(i)$ where $N(i)=\{j|(i,j)\in E(G)\}$.

To illustrate the above assumptions, consider two nodes $k,\ell\in V(G)$ where $\ell\in N(k)$ and $N(\ell)=\{k\}$. Then k knows that $\{k\} \subseteq N(\ell)$ but does not know $\{k\}=N(\ell)$ hence even in this simple case $INF(\ell,0) \not\subseteq INF(k,0)$.

Unless otherwise stated, we will use the following standard assumptions in our algorithms:

1) Partial reliability -- the network topology does not change during the execution time of an algorithm.
2) The network is synchronous -- it takes one unit of time to transmit a message along any edge.
3) The time to process a message at a node (including its queueing time) is negligible when compared with the time to transmit a message along an edge.

We now list the graph theoretical terms commonly used in the paper; the reader is referred to any standard graph theory text (for example [4]) for definitions of any additional term.

Given a graph $G = ((V(G)), E(G))$ we denote by:
$d(i,j)$ - the distance between $i,j\in V(G)$;
$r(i) = \max_{j\in V(G)}(d(i,j))$ - maximum distance between i and any other node;
$c(G) = \{j|r(j) = \min_{i\in V(G)} r(i)\}$ - set of centers of G;
$DIS(G,i) = \sum_{j\in V(G)} d(i,j)$ - sum of distances from a node i;

$m(G) = \{j \mid DIS(G,i) = \min_{i \in V(G)}(DIS(G,i))\}$ - set of medians of G;
$r(G) = r(i)$ where $i \in c(G)$ - radius of G;
$d(G) = \max_{i \in V(G)} r(i)$ - diameter of G.
In a tree T we denote by:
$<x,y>$ - the unique path between $x, y \in V(T)$;
diameter path - a longest path in T, i.e. any path with length $d(T)$;
$h(T)$ - height of T - (defined only for rooted trees) the length of the longest path from the root to any leaf.

3. TREE NETWORKS

3.1 A Basic Algorithm

In this section we present a basic algorithm which is a building block in the construction of distributed algorithms presented in this paper.

The purpose of this algorithm is to collect some relevant information from all nodes in the network and make it available to one or more nodes called *SATURATED* nodes. The *SATURATED* nodes can subsequently use this information to locate nodes with required properties (e.g. centers, medians, nodes with highest identifiers, etc.) or to determine properties of the network (e.g. diameter, number of nodes, etc.)

In the following sections we will use $S(t)$ to denote the set of nodes activated after t units of time from the start of the algorithm.

An arbitrary node I, called the initiator, activates all nodes in the network as follows:

(a) The initiator I sends a *FORWARD* message to all its neighbours.

(b) When a node is activated by a *FORWARD* message from one of its neighbours it sends a *FORWARD* message to all its other neighbours.

When a leaf (a node of degree one) is activated, it sends a *BACKWARD* message containing some information. The nature of this information may vary according to the context in which the basic algorithm is applied. For all other activated nodes which are not leaves, the algorithm proceeds as follows:

At each time instance t each node i checks which *BACKWARD* messages were received so far;

(a) If all but exactly one neighbour, say j, have already sent *BACKWARD* messages to i by this time, then i sends a *BACKWARD* message to j.

(b) If *BACKWARD* messages were received from all neighbours then i becomes *SATURATED*.

□

In the very special case when the initiator I is a leaf, then I sends the *BACKWARD* message to its only neighbour one time instance after the *FORWARD* message has been sent. In [17], a formal description of this algorithm is given.

We need some terminology in order to prove interesting properties of the basic algorithm. In a tree T every node $x \in V(T)$ with $|N(x)| = k$ is the root of k subtrees $\tau = T_\ell, \ldots, T_k$ such that for $i \neq j$ $T_i \cap T_j = \{x\}$, $\bigcup_{i=1}^{k} T_i = T$.

For a node $\ell \neq x$ we denote by $T_{\ell x}$ the unique tree in τ which contains ℓ.

Let $D = <a,b>$ be a diameter path in a tree T.

Lemma 3.1:

A node $x \in D$ can send a *BACKWARD* message only to a neighbour on D.

Lemma 3.2

For any diameter path D, a node $y \notin D$ cannot be *SATURATED*.

Theorem 3.1

Either one or two nodes become *SATURATED* in finite time, and every *SATURATED* node lies on all diameter paths.

It is easy to show that the number of messages exchanged until a *SATURATED* node is found is $2(n-1)$ if there is a single *SATURATED* node and $2n-1$ otherwise. This follows because exactly two messages will be transmitted along each edge with the exception of the edge between two *SATURATED* nodes on which three messages will be exchanged. We defer the analysis of the number of time instances to the next section.

3.2 Finding a Center in a Tree

As mentioned before, the knowledge of a center and of the radius of a network can be usefully employed in several algorithms to either minimize or bound the maximum transmission time and the volume of exchanged messages [16].

3.2.1 Basic Properties.

We present here some basic properties of trees which are used in our algorithm. Given a tree T:

Property 3.1:

If $d(T)$ is even there is a unique center, else there are two centers at distance one from each other.

Property 3.2:

$$r(T) = \left\lceil \frac{d(T)}{2} \right\rceil$$

Property 3.3:

Every diameter path contains all centers of T.

Property 3.4:

For a node $x \in T$ let j be the furthermost node from x in T i.e. $d(x,j) = r(x)$ then

$$d(j,c) \geq \left\lfloor \frac{d(T)}{2} \right\rfloor \quad \text{where } c \text{ is a center of } T.$$

3.2.2 The Algorithm TREE CENTER.

A distributed algorithm for finding the center in a tree was presented in [12]. To make the present paper self contained, we present a formal version of that algorithm. Furthermore, we will prove here that the algorithm is time optimal.

The algorithm is based on the observation that if a node x on the diameter path knows its distance from the two extreme points of this path, it can determine both the direction and the distance from itself to the closest center. This observation is formally stated in the following lemma:

Lemma 3.3:

If x is on a diameter path $D = \langle a,b \rangle$ and $d(x,a) = p_1$, $d(x,b) = p_2$ then

(a) c lies on $\langle x,a \rangle$ if $p_1 \geq p_2$ and on $\langle x,b \rangle$ otherwise.

(b) $d(x,c) = \left\lfloor \frac{|p_1 - p_2|}{2} \right\rfloor$

In order to find a center, we use the *BASIC* algorithm as described in the [17]. We use the *BACKWARD* messages to determine maximum distances from the *SATURATED* node. To achieve this, the leaves send *BACKWARD* messages with counters set to 1. These counters are incremented by one at each traversed node. The two largest counters that reach a *SATURATED* node s contain the distances from s to the two extremes of a diameter path say $D = <a,b>$. (See *Theorem 3.1*).

By *Lemma* 3.3 a *SATURATED* node knows $d(s,c)$ and whether $c \in <s,b>$. A formal description of the algorithm TREE-CENTER can be found in [17].

3.2.3 Analysis of the Algorithm.

In this section we state the correctness and optimality of algorithm TREE-CENTER.

Lemma 3.4

Algorithm TREE-CENTER correctly finds a center in a tree T.

Theorem 3.2

Any distributed algorithm for finding a center in a tree T from originator X requires at least $B(X) = r(X) + \lfloor \frac{d(T)}{2} \rfloor$ time units.

Theorem 3.3

Algorithm TREE-CENTER is time optimal.

It is easy to whow that the number of messages exchanged by algorithm TREE-CENTER is:

$$\begin{cases} 2(n-1)+r^* & \text{if there is a unique } \textit{STURATED} \text{ node} \\ 2(n-1)+r^*+1 & \text{if there are two } \textit{SATURATED} \text{ nodes} \end{cases}$$

where r^* is the minimum distance between a *SATURATED* node and a center. This follows because on each edge, two messages are sent (*FORWARD* and *BACKWARD*) whereas on every edge of $<s,c>$ a *SATURATED* message is added.

3.3 Finding a Median in a Tree

Another important node in a tree is a median, i.e. a node from which the average distance to all nodes in the tree is minimized [16]. In this section we show how to apply the BASIC algorithm in order to find the median.

3.3.1 Basic Properties.

Lemma 3.5

Let $x, y \in V(T)$ be connected by an edge then

$$DIS(T,x) = DIS(T,y) + |V(T_{xy})| - |V(T_{yx})|.$$

Let $\Delta(x,y) = DIS(T,x) - DIS(T,y)$ and $|V(T)| = N$

Corollary 3.1:

$$\Delta(x,y) = N + 2 - 2|V(T_{xy})| = -\Delta(y,x).$$

Lemma 3.6:

For $x \in T$ let $N(x) = \{1,2,\dots.j\}$ be the set of neighbours of x in T. There exists at most one vertex $i \in N(x)$ such that $\Delta(i,x) = N+2-2|V(T_{ix})| \leq 0$.

Lemma 3.7

A node $x \in T$ is a median of T if and only if for all $i \in N(x)$, $\Delta(x,1) \geq 0$.

Corollary 3.2

In a tree T with median m, for a node $y \in V(T)$ which is not a median there exists a unique $i \in N(y)$ with $\Delta(i,y) < 0$. In this case $i \in <y,m>$ and

$$|V(T_{iy})| > |V(T_{jy})| \text{ for } j \in N(y) - \{i\}.$$

Corollary 3.3:

There are at most two medians in a tree.

From all the above properties we can see that if a *SATURATED* node s knows all $\Delta(i,s)$ for $i \in N(s)$ it can determine whether s itself is a median (all $\Delta(i,s) \geq 0$) or if there exists $j \in N(s)$ such that $\Delta_{js} < 0$. In the latter case the median lies in the direction of j and s will report to j which will now become *SATURATED*.

3.3.2 The Algorithm TREE-MEDIAN. By the properties proved in the previous section it is clear that if a node x knows $DIS(T_{ix},x)$ and $|V(T_{ix})|$ for all $i \in N(x)$ it can compute $\Delta(i,x)$ and decide whether x is itself a median and if not so it can determine the direction of the median.

The purpose of our algorithm is to supply this information to the first *SATURATED* node from which the median is approached by a sequence of *SATURATED* nodes. This is done by providing two counters, m_1 and m_2 with each *BACKWARD* message sent. When a node x sends a *BACKWARD* message to y then

$$m_1 = DIS(T_{xy},x) - 1 \text{ and } m_2 = |V(T_{xy})| - 1.$$

Node y can process all *BACKWARD* messages received such that when y becomes *BACKWARD* it knows $DIS(T_{iy},y)$ and $|V(T_{iy})|$ for all but one neighbour to which it sends a *BACKWARD* message. This process continues until some node becomes *SATURATED* and the median can be found. The algorithm TREE-MEDIAN is the *BASIC* algorithm where the operations *SEND*, *PROCESS MESSAGE* and *SATURATED* are described in [17].

We now state a result analoguous to the one proved for finding the center in a tree.

Theorem 3.4:

Any distributed algorithm for finding the median of a tree T from initiator I, requires at least $\max_{x \in T}(d(I,x) + d(x,m))$ units of time.

Theorem 3.5:

Algorithm TREE-MEDIAN is time optimal for all trees with $|V(T)| \geq 3$.

4. GENERAL NETWORKS

4.1 Basic Algorithms

In many cases efficient distributed algorithms for finding certain properties of general networks, can be found by constructing certain spanning trees and then applying known algorithms which operate on trees [5,10,11]. Using this principle, we combine here the results of the previous section with a known algorithm for constructing spanning trees in order to find the center and the median of a general network.

Let us briefly describe a well known [9] algorithm that generates a spanning tree $SPT(x)$ rooted at a node $x \in V(G)$, such that the distance from x to any node $v \in V(G)$ in $SPT(x)$ is equal to $d(x,v)$ in G.

Algorithm SPANNING TREE

Step 1.

x sends a message *SP* to all $i \in N(x)$.

Step 2.

When a node v receives a message *SP* for the first time, it sends an *ACKNOWLEDGE* message along this edge and *SP* message along all other edges incident with v.

Step 3.

Every node v marks all the neighbours in G from which it received and to which it sent an *ACKNOWLEDGE* message. These are the neighbours of v in $SPT(x)$.

4.2 Finding Centers of General Networks

Our algorithm is based on some relations between the centers of a network and the centers of $SPT(x)$ for $x \in V(G)$. In general the radius of $SPT(x)$ for an arbitrary $x \in V(G)$ can be as large as $d(G)$ as shown in [17].

Therefore the simple heuristic of choosing a center in $c(SPT(x))$ of an arbitrary spanning tree $SPT(x)$ as an approximation may be as bad as choosing an arbitrary node. However we know that a center of G is also a center of its own spanning tree as proved in the following lemma.

Lemma 4.1:

If $x \in c(G)$ then $x \in c(SPT(x))$.

From this simple observation it follows that the only candidates of being centers of the graph are those nodes which are centers of their own spanning trees. We are now in a position to describe the algorithm GRAPH-CENTER starting from initiator I.

Algorithm GRAPH-CENTER

(a) Create a spanning tree $SPT(I)$

(b) Whenever a node x is activated in (a), it will initiate a process of finding the center of its own spanning tree $SPT(x)$.

(c) If node x finds that it is the center of $SPT(x)$ it is a candidate for being a center of G, it will then report $R(SPT(x))$ to I along the path $<x,I>$ in $SPT(I)$. This requires that in (a), each node marks its activator in $SPT(I)$.

(d) The initiator I will choose a center of G to be the node x which sent the minimum $r(SPT(x))$ within time $W = 4r(G) + 1$, and then will notify it. □

We now prove the correctness of the algorithm and show how I can compute W.

Lemma 4.2:

A message from a center x of G will arrive at I after t_x time units where $t_x \leq 4 \cdot r(G) + 1$.

This lemma guarantees that after $4 \cdot r(G) + 1$ time units, the initiator must have received a message from a center of G.

We now show how the initiator I can compute W without knowing $r(G)$. Initially I sets $W := \infty$, and its internal clock $t := 0$. Every time a report from a node x containing $r(SPT(x))$ is received, I updates its information about the minimum radius and the node that achieve this minimum.

```
if $4 \cdot r(SPT(x)) + 1 < W$ then
begin
        $W := 4 \cdot r(SPT(x)) + 1$;
        $tempc := x$;
end;
```

and, at every time instance I performs the following operations:

```
$t := t + 1$;
if $t = W$ then $tempc$ is the center.
```

Summarizing, the entire algorithm GRAPH-CENTER requires in total $W + d(I,c)$ time units where c is the closest center to I. This is bounded by $4 \cdot r(G) + 1 \leq W + d(I,c) \leq 5 \cdot r(G) + 1$.

An example of a graph in which our algorithm attains its upper bound is given in [17]. The lower bound will be achieved on any graph in which I is a center.

The number messages exchanged in this algorithm is fairly large. Each node has to participate in the construction and 'center finding' of n spanning trees. In each such spanning tree a node x will send at most $\deg(x)+1$ messages to find the center. In total this requires

$$\sum_{x \in V(G)} (deg(x)+1) = 2|E(G)| + |V(G)|$$

In the worst case every node has to report to I and this may require $DIS(SPT(I),I)$ additional messages. Finally, I has to notify the center; this requires at most $r(G)$ messages.

Since $DIS(SPT(I),I) \leq n^2$ we have an upper bound of

$$2n|E(G)| + 2n^2 + r(G)$$

on the number of exchanged messages where $n=|V(G)|$

4.3 Finding the Median in a Graph

An algorithm for finding a median can be devised along the same lines as the GRAPH CENTER algorithm, where now each node x reports to I the value of $DIST(SPT(x)x)$ (the sum of distances in $SPT(x)$ from x to any other node). Unlike GRAPH CENTER, the initiator I has to receive messages from all nodes in order to determine a median.

Algorithm GRAPH MEDIAN

(a) The initiator I creates a spanning tree $SPT(I)$.

(b) Whenever a node x is activated in (a) it will find $DIST(SPT(x),x)$ in $SPT(x)$ and report it to I.

(c) Initiator I chooses x whose report is minimum as the median and notifies x.□

In the worst case this algorithm requires 5d(G) units of time. It can also be shown that at most $2n|E(G)| + n^2 + d(G)$ messages will be exchanged.

The large number of messages echanged by algorithm GRAPH MEDIAN might motivate one to find approximate solutions.

In general, the simple heuristic of choosing the median m of an arbitrary tree $SPT(x)$ for some $x \in V(G)$, as an approximation to the correct median m of G can produce undesirble results. In fact, the ratio between $DIS(G,m_x)$ and $DIS(G,m)$ can be unbounded as shown in [17].

5. CONCLUDING REMARKS

Centers and medians play an important role in the construction of optimal broadcasting schemes and synchronization techniques in decentralized networks.

In this paper we have considered the problem of devising efficient distributed algorithms for finding a center and a median in a network. These algorithms can easily be extended to find all centers and all medians in a network.

The main contributions of this word are:

(a) Presenting a basic algorithm which can be applied for finding different properties of networks.

(b) Proving lower bounds on the time required to find centers and medians, in a tree network.

(c) Presenting time optimal algorithms for tree networks; the number of exchanged messages is linear in the number of nodes.

(d) Presenting algorithms for finding medians and centers for general networks. For a network with n nodes and e edges both algorithms run in $O(n)$ time units and require $O(ne + n^2)$ exchanged messages.

Throughout this paper, we have considered the case of only one node starting the process of finding the center or median of the network. This apparent limitation does not affect the generality of our algorithms. In fact, the case of several initiators can be reduced to the case of a simple initiator by means of a collision resolution mechanism based on a simple priority function [10].

In [7], lower bounds on the time needed to find centers medians and other important properties of general networks are presented. There seems to be a large gap between these lower bounds and the upper bounds presented here. Further research is needed to reduce this gap and also to improve the upper bounds on the number of messages exchanged.

It seems that in a general network, finding the exact center and median is a costly operation; therefore an important direction for research could be to devise algorithms which find a good approximate solution.

The problem of finding centers and medians in weighted networks is currently being investigated [7]. Preliminary results show that most of the algorithms presented here extend in a natural way to this more general case.

REFERENCES

[1] Abram, J.M., and Rhodes, I.B. A decentralized shortest path algorithm, Proc. 16th Allerton Conf. on Communication, Control, and Computing, 1978.

[2] Aburdene, M.F. Numbering the nodes of a graph by distributed algorithms, Proc. 17th Allerton Conf. on Communication, Control, and Computing, 1979.

[3] Angluin, D. Local and global properties in networks of processors, Proc. 12th ACM Symp. on Theory of Computing, 1980.

[4] Bondy, J.A., and Murty, U.S.R. Graph Theory With Applications, MacMillan, London, 1976.

[5] Chang, E.J. Decentralized algorithms in distributed systems. Tech. Rep. CCRG-103, Computer Systems Research Group, University of Toronto, 1979.

[6] Gallager, R.G., Humblet, P.A., and Spira, P.M. A distributed algorithm for minimum weight spanning trees, Res. Rep. LIDS-P-906-A, Mass. Inst. Tech., Laboratory for Information and Decision Systems, 1979.

[7] Korach, E., Rotem, D., and Santoro, N. Distributed algorithms for finding centers and medians in networks: the weighted case, in preparation.

[8] Lynch, N.A. Fast allocation of nearby resources in a distributed system, Proc. 12th ACM Symp. on Theory of Computing, 1980.

[9] Moore, E.F. The shortest path through a maze, Proc. Int. Symp. on Theory of Switching, 1959.

[10] Ramirez, R.J., and Santoro, N. Distributed control of updates in multiple-copy databases: a time optimal algorithm, Proc. 4th Berkeley Conf. on Distributed Data Management and Computer Networks, 1979.

[11] Romani, F. Cellular automata synchronization, Information Sciences, 10,3 (1976), pp. 299-318.

[12] Santoro, N. Determining topology information in distributed networks, Proc. 11th Southeastern Conf. on Combinatorics, Graph Theory and Computing, 1980.

[13] Slater, P.J., Hedetniemi, S.T., and Cockayne, E.J. Information dissemination in trees, Tech. Rep. CS-TR-78-11, Computer Science Department, University of Oregon, 1978.

[14] Tajibnapis, W.D. The design of a topology information maintenance scheme for a distributed computer network, Proc. ACM Conference, 1974.

[15] Tajibnapis, W.D. A correctness proof of a topology information maintenance protocol for a distributed computer network, CACM, 20,7 (July 1977) pp.477-485.

[16] Wall, D.W., and Owicki, S.S. Center-based broadcasting, to appear 1980.

[17] Korach, E., Rotem, D., and Santoro, N. Efficient distributed algorithms for finding centers and medians in networks, Research Rep. , Department of Computer Science, University of Waterloo, 1980.

OPTIMAL QUADRATIC STABILIZABILITY OF UNCERTAIN LINEAR SYSTEMS[†]

C.V. HOLLOT and B.R. BARMISH
Department of Electrical Engineering
University of Rochester
Rochester, New York 14627

Abstract: The problem of stabilizing uncertain linear systems is addressed in this paper. The uncertainty, $q(\cdot)$, which enters the system dynamics lacks a statistical description. Instead, the uncertainty is described via a set Q which serves as a bound for the admissible variations of $q(\cdot)$. The results given here differ from previous work in the following manner: We do not restrict the uncertain system parameters via the so-called "matching conditions;" nor do we enforce (possibly conservative) bounds on the uncertainty. Instead, the problem of stabilization is reformulated as a search for an "optimal" quadratic Lyapunov function. Such an approach leads to criteria which are both necessary and sufficient for quadratic stabilizability. We demonstrate the efficacy of this method via an example which cannot be handled via the theory developed to date.

I. INTRODUCTION

The study of uncertain dynamical system stabilization has been addressed in the recent literature. (See, for example, [1]-[14] and their bibliographies.) Under consideration are those systems for which statistical information pertaining to the uncertainty and/or disturbance is not available. Rather, we only claim a priori knowledge of the bounds on the region in which the disturbance lies. The objective is to provide stable system operation under all possible variations of the uncertainty within the given bounds. It has been the aim of the above mentioned studies to describe or define classes of uncertain systems for which this so-called guaranteed stability can be assured.

In this paper, we focus our attention on the theory and algorithms which enable one to achieve global asymptotic stability of an uncertain system via a quadratic Lyapunov function. Here, our goal is to extend and generalize the existing methods for guaranteeing stable operation of uncertain systems. Many of the results of [1]-[14] fall within the domain of the analysis given here. These recent approaches to the guaranteed stability problem may be divided into two classes:

(i) Those approaches (such as [1]-[10]) which seek a direct solution (control) via the imposition of certain "matching conditions." These conditions must be viewed as a priori assumptions which restrict the manner in which the uncertain parameters enter structurally into the state equations.

(ii) Approaches such as [11]-[14] which consider stabilization as a favorable by-product of a guaranteed cost performance problem. Such an approach is motivated naturally from an underlying optimal control problem.

The first class (i) above, provides one with direct means for determining whether an uncertain system is stabilizable and, if so, for synthesizing controls which accomplish the desired stable operation. For such situations, however, the class of uncertain systems to which the theory applies is limited by the a priori matching requirement. It is easy to construct systems of practical interest which cannot be handled

---

[†] This work was partially supported by the Rochester Gas and Electric Corp. and the U.S. Dept. of Energy under Contract ET-78-S-01-3390.

in this context -- but are nevertheless stabilizable. Such an example will be given in Section VI.

Alternatively, the second class (ii) of stabilization schemes often leads to stability of systems which are also required to satisfy other performance criteria. Although matching conditions are not explicitly required, these indirect methods are nevertheless limited in one important way: The uncertainty bounds for which the theory is applicable may be conservative. That is, it is possible to construct examples of stabilizable systems whose uncertainty bounds are too large to be handled via the approach described above. We also note that this comment applies to the so-called mis-match theory in [1].

Analogous to the first class (i) of stabilization methods, this paper establishes guaranteed stability results by exploiting a quadratic Lyapunov candidate

$$V(x) \overset{\Delta}{=} x'Px$$

where P is a $n\times n$ constant, positive-definite symmetric matrix. The main difference between this work and the preceeding work stems from the fact that we treat the entries of the Lyapunov P matrix as design variables which must satisfy certain constraints. Unlike the previous work, a successful selection of P is not contingent upon the satisfaction of matching conditions.[†] That is, we make no a priori assumptions about the manner in which the uncertainty enters structurally into the state equations. Instead, we show that it is possible to generate a set of inequalities which tell the designer whether or not the given system can be stabilized.[††] Hence, the guaranteed stability problem is viewed as a search in a parameter space of positive-definite symmetric matrices. If the search is successful, then the system is shown to be globally asymptotically stabilizable. We note that this method leads to conditions which are both necessary and sufficient for global asymptotic stability via a quadratic Lyapunov function.

Finally, if a system is found to be stabilizable by this method, it is of interest to construct a linear feedback control law which achieves the predicted stability property. This problem of controller synthesis will also be examined in this paper.

II. SYSTEM (SY) AND ASSUMPTIONS

The class of systems which we shall examine are linear and described by the state equation

$$\dot{x}(t) = A(q(t))x(t) + Bu(t); \quad t\varepsilon[0,+\infty) \qquad \text{(SY)}$$

where $x(t)\varepsilon R^n$ is the state; $u(t)\varepsilon R^m$ is the control and $q(t)\varepsilon Q\subset R^p$ is the time-varying model uncertainty which is restricted to a prespecified bounding set Q. In addition, B is taken to be a constant[†††] $n\times m$ matrix and $A(\cdot)$ is an $n\times n$ dimensional matrix function defined on the set Q. Hence, for fixed $q\varepsilon Q$, $A(q)$ is the model matrix which results. Throughout the remainder of this paper, the following assumptions are taken as standard.

[†] generalized matching conditions in the case of [10].

[††] depending on whether or not a feasible point can be found.

[†††] This assumption can be weakened; i.e., one can permit B to depend on the uncertainty q. Consequently if B(q) satisfies certain "matching conditions" (see [6]), then the analysis to follow can be easily modified.

Assumption 1: $A(\cdot)$ is continuous in its argument.

Assumption 2: The bounding set $Q \subset R^p$ is compact.

Assumption 3: The uncertainty $q(\cdot)$: $[0,+\infty) \to Q$ is required to be Lebesgue measurable. We say that such $q(\cdot)$'s are admissible uncertainties.

Assumption 4: The number of controls m is less than[†] the number of system states n. Furthermore, it is assumed that $\operatorname{rank} B = m$.

Remark: The preceding assumption is made without loss of generality since redundant inputs can always be eliminated.

Remarks about (SY): Given the system (SY), the so-called guaranteed stability problem involves constructing a controller which renders the system uniformly and asymptotically stable. This stability must be guaranteed regardless of the time-varying nature of the disturbance $q(\cdot)$. It is only assumed that the values $q(t)$ lie within the set Q. In order to achieve this goal, one usually imposes additional restrictions on the system (SY). For example, in Leitmann's work [6], the system matrix $A(q)$ must satisfy certain matching conditions; i.e., the uncertain matrix $A(q)$ must be decomposable as

$$A(q) = A_0 + BD(q) \tag{2.1}$$

where $D(\cdot):Q \to R^{m \times n}$ is a matrix function which is continuous in its argument and (A_0, B) is a stabilizable pair. More recently, Barmish and Leitmann [1], have relaxed these matching conditions. Rather than emphasizing the need for an exact matching criteria, they have introduced certain scalars which are termed the measure of mis-match and the mis-match threshold. Guaranteed stability is achievable when the computed value of the measure of mis-match is less than the mis-match threshold. Thorp and Barmish [10], have also introduced even less restrictive matching conditions on $A(q)$ which allow more freedom in the uncertainty structure. Their theory likewise guarantees stability for a class of uncertain systems. Finally, we mention the more recent work of Molander [9] wherein the structure of the uncertainty is restricted by certain subspace relationships.

In contrast to the existing literature, we shall not make any matching assumptions nor shall we place possibly conservative bounds on the uncertainty. Instead, we shall take $A(q)$ and Q as given and provide a "yes" or "no" answer as to whether (SY) can be stabilized via a quadratic Lyapunov function. The conditions which we develop will be both necessary and sufficient.

III. GLOBAL QUADRATIC STABILIZABILITY AND A TESTING MATRIX $\pi(\cdot,\cdot)$

Before presenting the main results of this paper, we shall develop two concepts.

Global Quadratic Stabilizability: Consider the uncertain dynamical system (SY). Then (SY) is said to be globally quadratically stabilizable if there exists:

[†] For the special case of n=m, B has full rank and is therefore invertible. It is easy to show that the control $u=-\gamma B^{-1}x$, $\gamma > 0$, stabilizes (SY) for γ sufficiently large.

1) a positive-definite symmetric matrix P,

2) a feedback control law $p(\cdot): R^n \to R^m$, and

3) a continuous scalar function $c_0(\|x\|)$ with $c_0(0) = 0$,

such that the following conditions hold:

i) Given any admissible uncertainty $q(\cdot)$, solutions to the resulting non-linear system

$$\dot{x}(t) = A(q(t))x(t) + Bp(x(t)) \tag{3.1}$$

exist and are continuable over $t \varepsilon [0,+\infty)$.

ii) Given the Lyapunov candidate $V(x) \overset{\Delta}{=} x'Px$ and any admissible uncertainty $q(\cdot)$, it follows that the Lyapunov derivative

$$L(x,t) \overset{\Delta}{=} (\text{grad}_x V(x))'\dot{x}(t) \leq c_0(\|x\|) < 0$$

for all $x \varepsilon R^n$ and for all $t \varepsilon [0,+\infty)$.

Remark: Employing (SY), a simple calculation yields

$$L(x,t) = x'(A'(q(t))P + PA(q(t)))x + 2x'PBp(x). \tag{3.2}$$

Construction of a Testing Matrix $\pi(\cdot,\cdot)$: Paramount to our stability criterion is a certain matrix $\pi(S,q)$. We now describe its construction: For the case when $m < n$, consider

Step 1: Let $\tilde{b}_i \varepsilon R^n$, $i=1,2,\ldots,(n-m)$ be a set of basis vectors for the nullspace of B'.

Step 2: Given any $n \times n$ positive-definite symmetric matrix S one forms an intermediate matrix

$$M(S,q) \overset{\Delta}{=} SA'(q) + A(q)S. \tag{3.3}$$

Step 3: Finally, we construct the $(n-m) \times (n-m)$ matrix $\pi(S,q)$ having $(j,k)^{th}$ entry

$$\tilde{b}_j' M(S,q)\tilde{b}_k. \tag{3.4}$$

IV. GUARANTEED STABILITY

We are now ready to state a guaranteed stability theorem.

Theorem 4.1 (See Appendix A for proof): The dynamical system (SY) is globally quadratically stabilizable if and only if there exists a positive-definite symmetric matrix S such that $\pi(S,q)$ is negative-definite for all $q \varepsilon Q$.

Computation: The preceding theorem immediately suggests an approach to numerical computation. Letting $\Delta_i(S,q)$ denote the i^{th} principal minor of $-\pi(S,q)$, we observe that:

i) By the theorem of Sylvester $\Delta_i(S,q) > 0$ for $i=1,2,\ldots,(n-m)$ is necessary and sufficient for the negative-definiteness of $\pi(S,q)$.

ii) $\Delta_i(S,q) \geq \min_{q \varepsilon Q} \Delta_i(S,q)$ for all $q \varepsilon Q$.

A reformulation for quadratic stabilizability immediately follows.

Corollary 4.2: (SY) *is globally quadratically stabilizable if and only if there exists a positive-definite symmetric matrix* S *such that*

$$\min_{q \in Q} \Delta_i(S,q) > 0; \quad i = 1,2,\ldots,n-m. \tag{4.1}$$

Consequently, with the aid of this corollary, we see that determination of an appropriate matrix S is tantamount to finding a feasible point for a system of inequalities. For particular examples, this can be accomplished with the aid of algorithms supplied from the theory of finite-dimensional mathematical programming (see, for example,[15] and [16]). We shall illustrate the technique suggested above via a numerical example.

Previous Research: When one is fortunate enough to have a system (SY) which satisfies the matching conditions of [6], the determination of the matrix S is greatly facilitated. For example, if $A(q) = A_0 + BD(q)$ as in (2.1), one would proceed as follows:

a) Construct a linear feedback matrix K_0 such that $\bar{A}_0 \overset{\Delta}{=} A_0 + BK_0$ has all its eigenvalues in the strict left half plane. (This is possible because (A_0,B) is a stabilizable pair.)

b) Solve the Lyapunov equation $\bar{A}_0'P + P\bar{A}_0 = -I$ where I denotes the identity matrix.

c) Using the solution matrix P obtained in (b), simply set $S = P^{-1}$.

We also note that a simple "recipe" for constructing S is possible when the system (SY) satisfies the generalized matching conditions of [10]; i.e., one constructs a Lyapunov matrix P as suggested in [1], [4] and [6] and once again sets $S = P^{-1}$.

Linear Feedback: When the conditions of Theorem 4.1 are met, it is of interest to know whether or not the control $p(x)$ can be realized as a linear mapping on the state. In other words, it is often desirable to achieve guaranteed stability with a controller $p_\ell(x) = Kx$ where K is a constant m×n matrix. The following corollary tells us that the existence of a nonlinear controller $p(x)$ implies the existence of an appropriate linear controller $p_\ell(x)$.

Corollary 4.3 (See [21] for proof): *Suppose that the inequalities in* (4.1) *are satisfied by some positive-definite symmetric matrix* S. *Then* (SY) *is globally quadratically stabilizable via a linear feedback control law of the form*

$$p_\ell(x) = -\gamma B'S^{-1}x \tag{4.2}$$

where γ *is a positive scalar.*

The preceding corollary immediately suggests a numerical approach for determining the positive scalar γ of equation (4.2). Suppose that a matrix P has already been found satisfying the conditions of Theorem 4.1. Hence, one can define

$$\bar{\pi}(\gamma,q) \overset{\Delta}{=} A'(q)P + PA(q) - 2\gamma PBB'P \tag{4.3}$$

and let $\bar{\Delta}_i(\gamma,q)$ denote the i^{th} principal minor of $-\bar{\pi}(\gamma,q)$. As done in Corollary 4.2, we can reformulate the criterion for quadratic stabilizability via linear feedback in the following manner:

Corollary 4.4: Suppose that (SY) is globally quadratically stabilizable via a linear feedback control law of the form of equation (4.2). Then there exists a positive scalar γ, such that

$$\min_{q\in Q} \overline{\Delta}_i(\gamma,q) > 0, \quad i=1,2,\ldots,(n-m). \tag{4.4}$$

Hence, we conclude that the determination of γ involves a one-dimensional search. We shall illustrate this technique via a numerical example.

In light of the preceding corollaries, we conclude that the system designer is faced with a choice between linear and nonlinear control. We shall not dwell on this problem here, but instead focus on the computation of the matrix S. It is important to note that the problem of finding S is independent of one's choice of linear versus nonlinear control.

V. SPECIAL CASE: LINEAR UNCERTAINTIES WITH POLYHEDRAL CONSTRAINTS

In this section, we specialize our preceding results to systems described by linear uncertainties in $A(q)$ and polyhedral constraints describing Q. We note that the systems and uncertainties considered in [11] and [14] fall within this class.

Definition: The dynamical system (SY) is said to be described by *linear uncertainties and polyhedral constraints* if Q is a polytype and $A(q)$ can be expressed as

$$A(q) = A_0 + \sum_{i=1}^{p} q_i A_i \tag{5.1}$$

where $A_0, A_1, \ldots, A_p$ are constant $n\times n$ matrices and $q_1, q_2, \ldots, q_p$ are the components of the uncertain vector q. The vertices of the polytype Q will be denoted by $\tilde{q}^1, \tilde{q}^2, \ldots, \tilde{q}^\ell$.

Note that the conditions of the preceding definition imply the satisfaction of standard Assumptions 1 and 2.

Theorem 5.1 (See [21] for proof): *Suppose that the dynamical system (SY) is described by linear uncertainties and polyhedral constraints. Then (SY) is globally quadratically stabilizable if and only if there exists a positive-definite symmetric matrix S such that $\pi(S,\tilde{q}^i)$ is negative-definite for $i=1,2,\ldots,\ell$.*

VI. EXAMPLE

Consider the system described by

$$A(q) = \begin{bmatrix} -1 & 0 & q \\ q & -2 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \quad |q| \le \overline{q} = 5. \tag{6.1}$$

Since this system has uncertainty in the (1,3) entry of $A(q)$, it is apparent that the matching conditions of [4], [6] and [10] are not satisfied. Furthermore, it can also be easily verified that the measure of mis-match defined in [1] exceeds the so-called mis-match threshold. Hence, the stabilization theories given in [1], [4], [6] and [10] are not applicable. Finally, one might consider handling this system via the theory of Guaranteed Cost Control (see [11]). Such an approach requires creation of a quadratic performance index. This leads to the problem of solving a Riccati equation which is augmented by an additional term to account for the uncertainty. Using unit weighting matrices, we employed the iterative approach of Kleinman [17] and deduced that stationary solutions do not exist for $\bar{q}=5$. Therefore, it was concluded that one could not assure stability with such a performance index.

Setting up the inequalities prescribed by Corollary 4.2, one obtains constraints on s_{ij}, the entries of S. They are

$$-s_{11}+\bar{q}s_{13}<0;\ 4(-s_{11}+s_{13}\bar{q})(2s_{22}-s_{12}\bar{q})+(3s_{12}-s_{11}\bar{q}-s_{23}\bar{q})^2<0;$$

$$-s_{11}-\bar{q}s_{13}<0;\ s_{11}>0;\ s_{11}s_{22}-s_{12}^2>0;$$

$$-4(s_{11}+s_{13}\bar{q})(2s_{22}+s_{21}\bar{q})+(3s_{21}+s_{11}\bar{q}+s_{23}\bar{q})^2<0;$$

$$s_{11}s_{22}s_{33}+2s_{12}s_{13}s_{23}-s_{13}^2s_{22}-s_{12}^2s_{33}-s_{23}^2s_{11}>0.$$

It is clear that this set of constraints can be satisfied using a diagonal matrix whose entries are $s_{11}=1$, $s_{12}=10$ and $s_{33}=1$. By then applying Corollary 4.4, one solves for the scalar $\gamma>0$ and generates a feedback gain matrix $K=[0\ 0\ 10.1]'$. Such an approach, however, does not take advantage of the fact that the s_{ij} satisfying the preceding inequalities are highly non-unique. This non-uniqueness enables us to improve on the preceding solution; i.e., one can introduce a performance index $J(S,\gamma)$ which can be used to reflect secondary considerations -- over and above stability. For example, let J measure the norm of the resultant feedback gain matrix K (note K depends on S and γ). Then minimizing $J^{\dagger}$ with respect to the entries of S and γ subject to the constraints imposed by Corollary 4.4, one obtains a solution $s_{11}=2.16$, $s_{12}=s_{21}=.018$, $s_{13}=s_{31}=.0007$, $s_{22}=9.66$, $s_{23}=s_{23}=-.818$, $s_{33}=1.18$ and $\gamma=6.94$. The resultant feedback gain matrix $K=[-.007\ \ .541\ \ 6.33]$ can then be used to achieve the desired stability.

APPENDIX A - PROOF OF THEOREM 4.1

Proof of Theorem 4.1: (Necessity) Suppose that the system (SY) is globally quadratically stabilizable. Consequently, there exists a positive-definite symmetric matrix P, a control law $p(\cdot):R^n\rightarrow R^m$ and a continuous scalar function $c_0(\|x\|)$ such that given any admissible uncertainty $q(\cdot)$,

$$L(x,t)\le c_0(\|x\|);\ L(x,t)\le c_0(\|x\|) \tag{A.1}$$

$$c_0(\|x\|)<0 \tag{A.2}$$

for all $x\varepsilon R^n$ and all $t\varepsilon[0,+\infty)$. The objective is to show that there exists a positive-definite symmetric matrix S such that $\pi(S,q)$ is negative-definite for all admissible $q\varepsilon Q$.

Recalling that the vectors $\tilde{b}_i$ are basis vectors for the nullspace of B', it is easily shown that the vectors

$$v_i \triangleq P^{-1}\tilde{b}_i,\quad i=1,2,\ldots,(n-m) \tag{A.3}$$

[†] This minimization was accomplished using a penalty function method which we implemented via the IBM subroutine ZXMIN.

constitutes a basis for the nullspace of B'P.

Now, let $q(\cdot)$ be any admissible uncertainty. Since (A.1) holds for all $x \varepsilon R^n$, it certainly holds for all vectors x which are linear combinations of the v_i above. Hence, if

$$x = \sum_{i=1}^{n-m} \alpha_i v_i ; \quad \alpha_i \varepsilon R, \tag{A.4}$$

one can define an equivalent expression for $L(x,t)$ which holds when $B'Px=0$. Namely, let

$$\tilde{L}(\alpha,t) \triangleq L\left(\sum_{i=1}^{n-m} \alpha_i v_i , t\right). \tag{A.5}$$

Then from (A.1) and (A.4), it is clear that for all $\alpha \varepsilon R^{n-m}$ and all $t \varepsilon [0,+\infty)$

$$\tilde{L}(\alpha,t) \le c_0\left(\left\| \sum_{i=1}^{n-m} \alpha_i v_i \right\|\right) < 0. \tag{A.6}$$

Recalling the expression for the Lyapunov derivative given by equation (3.2) and using (A.4), we form

$$\begin{aligned} \tilde{L}(\alpha,t) = {} & \left(\sum_{j=1}^{n-m} \alpha_j v_j \right)' (A'(q(t)P + PA(q(t))) \left(\sum_{k=1}^{n-m} \alpha_k v_k \right) \\ & + 2\left(\sum_{i=1}^{n-m} \alpha_i v_i \right)' PBp(x) \le c_0\left(\left\| \sum_{i=1}^{n-m} \alpha_i v_i \right\|\right) < 0. \end{aligned} \tag{A.7}$$

Noting that $v_i' PBp(x) = 0$ for $i = 1,2,\ldots,(n-m)$, one uses (A.3) to obtain

$$\begin{aligned} \tilde{L}(\alpha,t) = {} & \sum_{j=1}^{n-m} \sum_{k=1}^{n-m} \alpha_j \alpha_k \tilde{b}_j' (A(q(t))P^{-1} + P^{-1}A'(q(t)))\tilde{b}_k \\ & \le c_0\left(\left\| \sum_{i=1}^{n-m} \alpha_i v_i \right\|\right) < 0. \end{aligned} \tag{A.8}$$

Since the preceding inequality holds for all admissible $q(\cdot)$, it must hold when $q(t) \equiv q$ is constant. In addition, by selecting $S \triangleq P^{-1}$, (A.8) then requires that

$$\sum_{j=1}^{n-m} \sum_{k=1}^{n-m} \alpha_j \alpha_k \tilde{b}_j' (A(q)S + SA'(q))\tilde{b}_k \le c_0\left(\left\| \sum_{i=1}^{n-m} \alpha_i v_i \right\|\right) < 0 \tag{A.9}$$

must hold for all $\alpha \varepsilon R^{n-m}$ and all $q \varepsilon Q$. Furthermore, the summand of (A.9) is precisely the (j,k)th element of $\pi(S,q)$, (see equation (3.3)) so that

$$\alpha' \pi(S,q) \alpha \le c_0\left(\left\| \sum_{i=1}^{n-m} \alpha_i v_i \right\|\right) < 0 \tag{A.10}$$

must hold for all $\alpha \varepsilon R^{n-m}$ and all $q \varepsilon Q$. From (A.10) we conclude that $\pi(S,q)$ is negative-definite for all $q \varepsilon Q$.

(Sufficiency) Now suppose that there exists a positive-definite symmetric matrix S such that $\pi(S,q)$ is negative-definite for all $q \varepsilon Q$. We now define

$$P \triangleq S^{-1} \tag{A.11}$$

and a controller $p(\cdot):R^n \to R^m$ which is a continuous selection[†] from the set-valued mapping

$$\Gamma(x) \overset{\Delta}{=} \{p\varepsilon R^m: \sup_{q\varepsilon Q} 2x'PA(q)x + 2x'PBp(x) \le \frac{1}{2}\tilde{\lambda}\|x\|^2 \tag{A.12}$$

where

$$\tilde{\lambda} \overset{\Delta}{=} \max_{q\varepsilon Q} \frac{\lambda_{\max}[\pi(S,q)]}{\lambda_{\min}[V]} \tag{A.13}$$

and V is the $(n-m)\times(n-m)$ matrix whose $(j,k)^{th}$ entry is $v_i'v_j$ and $\lambda_{\max(\min)}[\cdot]$ denotes the operation of taking the largest (smallest) eigenvalue. In accordance with [18], the existence of such a continuous selection is guaranteed if the set $\Gamma(x)$ is non-emptry, closed and convex for all $x\varepsilon R^n$ and in addition, $\Gamma(\cdot)$ must be lower semicontinuous as a set-valued mapping. In the more expanded version of this paper [21], it is shown that $\Gamma(\cdot)$ does indeed satisfy these requirements. To complete our construction we define

$$c_0(\|x\|) \overset{\Delta}{=} \frac{1}{2}\tilde{\lambda}\|x\|^2.$$

Remarks: 1) The linear independence of the v_i's assure that V is positive-definite. Moreover, if

$$x = \sum_{i=1}^{n-m} \alpha_i v_i,$$

then it is easily shown that

$$\|x\|^2 \ge \lambda_{\min}[V]\|\alpha\|^2. \tag{A.14}$$

2) In (A.13), the maximum defining $\tilde{\lambda}$ is attained by some vector $\tilde{q}\varepsilon Q$. This follows because $\lambda_{\max}[\cdot]$, and $\pi(S,\cdot)$ are continuous and Q is compact. Hence, it follows that $\tilde{\lambda}$ must be negative as a consequence of the negative-definiteness of $\pi(S,\tilde{q})$ and positive-definiteness of V. We now recognize that $c_0(\|x\|) < 0$ for all non-zero $x\varepsilon R^n$.

To complete the proof of global quadratic stabilizability we must demonstrate the conditions (i) and (ii) of the global quadratic stabilizability definition are satisfied.

Let $q(\cdot)$ be any admissible uncertainty. Then the right hand side of (3.1) with continuous control $p(\cdot)$ selected from (A.12), is continuous in x and Lebesgue measurable in t. Hence, given initial condition (x_0,t_0), there exists a solution $x(\cdot):[t_0,t_0+\Delta t]\to R^n$, $x(t_0)=x_0$ for some $\Delta t>0$ (see, for example, [6]). Then under control law $p(\cdot)\varepsilon\Gamma(\cdot)$, the resulting Lyapunov derivative in (3.2) is written

$$L(x,t) = 2x'PA(q(t))x + 2x'PBp(x). \tag{A.15}$$

Now, let the pair $(x,t)\varepsilon R^n\times[0,+\infty)$ be chosen arbitrarily. Then from equations (A.12) and (A.15)

[†] If X and Y are topological spaces and ϕ a function from X to subsets of Y, then a continuous selection for ϕ is a continuous $f:X\to Y$ such that $f(x)\varepsilon\phi(x)$ for every x in X (see [18]).

$$L(x,t) \leq \max_{q \in Q} 2x'PA(q)x + 2x'PBp(x) \leq \frac{1}{2}\tilde{\lambda}\|x\|^2 = c_0(\|x\|) < 0.$$

Having shown that there exists a solution at every $(x_0,t_0) \in R^n \times [0,+\infty)$, it follows from the argument above that $L(x,t)$ is strictly decreasing for all pairs (x,t). Hence, no such solution has finite escape time and may be continued indefinitely over $[0,+\infty)$. □

REFERENCES

[1] Barmish, B.R. and Leitmann, G., On Ultimate Boundedness Control of Uncertain Systems in the Absence of Matching Assumptions, to appear.

[2] Barmish, B.R. and Petersen, I.R., Non-linear versus Linear Control for a Class of Ultimate Boundedness Problems With Uncertainty, to appear.

[3] Corless, M. and Leitmann, G., Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems, to appear.

[4] Gutman, S., Uncertain Dynamical Systems-A Lyapunov Min-Max Approach, IEEE Transactions on Automatic Control, Vol.AC-24, No.3, June 1979.

[5] Leitmann, G., Guaranteed Ultimate Boundedness for a Class of Uncertain Linear Dynamical Systems, Proceedings of the Third Kingston Conference on Differential Games and Control Theory, Marcel Dekker, NY, 1979.

[6] _____, On the Efficacy of Nonlinear Control in Uncertain Linear Systems, Proceedings of the 1980 Winter Meeting of the ASME, to appear.

[7] _____, Stabilization of Dynamical Systems Under Bounded Input Disturbances and Parameter Uncertainty, Second Kingston Conf. on Differential Games and Control Theory, Marcel Dekker, Inc. NY, 1976.

[8] _____, Guaranteed Asymptotic Stability for Some Linear Systems With Bounded Uncertainties, Journal of Dynamic Systems, Measurement and Control, Vol.101, No.3, 1979.

[9] Molander, P., Stabilization of Uncertain Systems, Report LUTFD2/(TRFT-1020)/1-111/(1979). Lund Institute of Technology, August, 1979.

[10] Thorp. J.S. and Barmish, B.R., On Guranteed Stability of Uncertain Linear Systems Via Linear Control, Journal of Optimization Theory and Applications, in press.

[11] Chang, S.S.L. and Peng, T.K.C., Adaptive Guaranteed Cost Control of Systems with Uncertain Parameters, IEEE Transactions on Automatic Control, AC-17, August 1972, pp. 474-483.

[12] Peng, T.K.C., Invariance and Stability for Bounded Uncertain Systems, SIAM J. Control, Vol.10, No.4, November 1972, pp. 679-690.

[13] Vinkler, A. and Wood L.J., A Comparison of Several Techniques for Designing Controllers of Uncertain Dynamic Systems, Proceedings of the IEEE Conference on Decision and Control, 1978.

[14] Vinkler, A. and Wood, L.J., Multistep Guaranteed Cost Control of Linear Systems with Uncertain Parameters, Journal of Guidance, Vol.2, No.6, pp. 449-456.

[15] Luenberger, D.G., Optimization by Vector Space Methods, Wiley, NY, 1969.

[16] Polak, E., Computational Methods in Optimization, Academic Press, 1971.

[17] Kleinman, D.L., On an Iterative Technique for Riccati-Equation Computation, IEEE Transactions on Automatic Control, Vol. AC-13, Feb. 1968, pp. 114-115.

[18] Michael, E., Selected Selection Theorems, American Math. Monthly, Vol.63, 1956, pp. 233-236.

[19] Zakian, V. and Al-Naib, U., Design of Dynamical and Control Systems by the Method of Inequalities, Proceedings IEE, Vol.120, No.11, Nov. 1973, pp. 1421-1427.

[20] Chen, C.T., Introduction to Linear System Theory, Holt, Rinhart and Winston, Inc., 1970.

[21] Hollot, C.V. and Barmish, B.R., Stabilization of Uncertain Linear Systems via a Variable Lyapunov Function, in preparation.

STABILITY OF SELF PROTECTING SYSTEMS

JAMES S. THORP, ROBERT J. THOMAS & JOHN W. McCLURE

School of Electrical Engineering
Cornell University
Ithaca, NY 14853

ABSTRACT

The impact of line protections equipment on the stability of large-scale electric power systems is considered. In order to capture inadvertent relay operations caused by carrier-blocking failures the system is modeled as an uncertain dynamic system. Worst case disturbances are obtained by examining the Zubov function for the post-fault system. A new technique for obtaining approximate Zubov functions based on results in the stability of sets of matrices is suggested.

INTRODUCTION

It is acknowledged that protection equipment plays an important role in determining the dynamic behavior of large-scale electric power systems. Indeed both the northeast blackout of 1965 and the New York City outage of July 1977 can be argued to have resulted from mis-set or mis-operating protection equipment. Long term dynamic simulations include the functions of the protection equipment to accurately represent the evolving behavior of the system. It must be observed, however, that simulations that represent the proper operation of the protection equipment capture only a fraction of the events that actually occur on the system. The following event is described in [1].

> "The incident occurred on a Saturday morning on the AEP system. A lightning arrester failure on the Gavin Marysville 765kv circuit, led to a permanent fault. (See Figure 1) On the AEP system, the standard reclosing procedures are as follows: A high Speed Reclosing without check synchronization in about 30 cycles (at 765kv). There is an attempt to reclose the line automatically at 10 seconds with check-synchronizing. Another attempt at reclosing is made in about 60 seconds, and if the fault is found to be permanent the line is locked out. It can only be reclosed manually after the lockout. When the initial fault led to a trip, High Speed Reclose was attempted at 10 cycles and the line tripped again. The other end of the Gavin Mountaineer line opened because of a hole in the blocking carrier. The Mountaineer station is the site of a new generating station; and an operating policy prohibits HSR when the plant has only two outlets. This policy wasin force at Mountaineer although as yet there was no generation at Mountaineer. Thus there was no HSR at the Mountaineer end after the false trip. The auto-reclose into the fault occured in about 10 seconds. At this time the breakers tripped, but a hole in the carrier on the Gavin-Amos line tripped

---

This work supported by the U.S. Department of Energy under Contract No. ET-78-S-01-3390.

> this line falsely. When this line tripped the 1300 MW unit at Gavin was isolated from the system. In about 9 cycles after the unit was thus 'trapped',

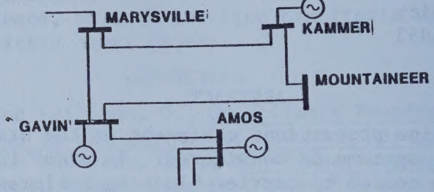


Figure 1
Incident at Gavin in AEP, 1979

> the Mountaineer line synchronized it again by its 10 second auto-reclose. However, at this time, the breaker failure relay at Mountaineer mis-operated because of a previously undetected wiring error. The Mountaineer line reopened, and by a transfer trip also tripped and locked out its Gavin end. The unit was therefore trapped again for about 8 cycles, at which time the High Speed Reclosing of the Amos line resynchronized it. Throughout these 9 and 8 cycles unit isolation and resynchronization no ill effects on the unit were observed. Since all this occurred at about 4 AM on a Saturday morning, nothing of consequence was felt on the system. There were two other incidents related to this event. One was a false trip of the Kammer Marysville circuit and the other a transformer differential due to an undetected shorted CT secondary."

Although this is a rare event, the holes in the blocking carrier are not as rare as might be imagined. Of 303 interruptions studied in [2] as many as 29% can be attributed to carrier-blocking failures. It is argued that "inadvertent relay operations which cause additional line tripouts are so numerous that a reclosing philosophy cannot be developed without considering the adverse effects of unwanted interruptions." [2]

The purpose of this paper is to consider the stability problems associated with these inadvertent line tripouts. It will be assumed that there is an initiating event such as the lightning arrester failure in the AEP example. It is also reasonable to assume that there is an ultimate post-fault equilibrium. (e.g. the Gavin Marysville line open and all other lines in service) Given the unpredictable nature of the carrier-blocking failures we will model the lines in a way that allows a worst case analysis. That is, we will question whether there is a sequence of inadvertent openings and reclosures that produce an unstable system.

FORMULATION

Consider the system

$$\mathring{x} = f(x,q) \qquad (1)$$

where $x(t) \varepsilon R^n$ is the state, $q(t) \varepsilon Q \subset R^\ell$ where ℓ is the number of lines. We take the equilibrium for the post-fault system to be x=0 and

$$q_j = 0 \;\Rightarrow\; \text{line } j \text{ in service}$$

$$q_j = 1 \;\Rightarrow\; \text{line } j \text{ open}$$

In other words, the system

$$\overset{\circ}{x} = f(x,0) = f(x) \tag{2}$$

describes the post-fault stability problem that would be studied without considering inadvertent openings due to carrier blocking failures. It is reasonable to assume that the inadvertent openings are limited to a large but finite time, T. The issue then is to determine whether at time T the state x(T) resulting from (1) is outside the domain of attraction of the origin of system (2). In a practical sense some additional restrictions must be placed on the bounding set Q. If all of the lines are allowed to be open for the entire interval then the system will surely be unstable. In order to produce a tractable problem that includes reasonably possible worst case events define

$$Q_p = \{q\text{: exactly } p \text{ lines are open}\}$$

A further restriction on q might be to permit q(t) to change only every ΔT secs where ΔT corresponds to the high speed reclose interval.

The system (1) is a restricted from of an uncertain dynamic system [3]. We would like to guarantee the stability of (1) under worst case disturbances i.e., for the "worst" $q(t) \in Q_p$. Some criterion must be used to establish the "worst" q(t). Ideally we would seek a Lyapunov-like function, V(x), for the system (2) with the following properties

V(x) Positive definite

$V(x) < v^*$ inside the domain of attraction of the origin

$V(x) > v^*$ outside the domain of attraction of the origin

where v^* is any constant.
If such a V(x) were found, then the worst q(t) would be formed as

$$\max_{q \in Q_p} V_x^T f(x,q) = V_x^T f(x,q^*)$$

and the solution of

$$\overset{\circ}{x} = f(x,q^*) \; ; \; x(0) = x_0$$

at the T denoted $x^*(T)$ could be used to determine stability. That is, the system is stable if

$$V(x^*(T)) < V^*$$

ZUBOV FUNCTIONS

The ideal Lyapunov function of the previous section has been suggested by Zubov [4] - [6]. The importance of Zubov's Method in the analysis of transient stability of power systems has been recognized by a number of authors [8]-[10]. Unlike conventional Lyapunov methods, Zubov's technique permits precise determination of the boundary of the domain of attraction of a dynamic system. For the system

$$\overset{\circ}{x} = f(x)$$

with stable equilibrium at the origin, Zubov gives a relationshop between a positive definite Lyapunov function, V(x), and the exact domain of attraction. In particular, V(x) satisfies the partial differential equation

$$\overset{o}{V}(x) = \phi(x) \quad (V(x) -1) \sqrt{1+ \|f(x)\|^2} \tag{3}$$

where $\phi(\underline{x})$ is positive definite or semi-definite and can be selected to influence convergence of numerical solutions. In the interior of the domain of attraction V(x) <1 and on the boundary V(x) = 1. The factor $\sqrt{1 + \|f(x)\|^2}$ in (3) can be omitted if the solutions to (2) are bounded for all t. The technique is applicable even when the domain is not bounded. An example is given in Hahn [5, p. 79] where the boundary of the domain is given by the curve $x_1 x_2 = 1$.

Unfortunately the Zubov technique is limited by the difficulty in obtaining a numerical solution to (3). Zubov suggested a power series expansion of V where sets of linear equations were solved for unknown coefficients. The series converges irregularly [6] and in the discussion of [8], it is shown that a 14th order series expansion for a system with eight state variable would require the soltuion of a matrix equation of the size 116,280. Even the applicaitons of the generalized Zubov method, due to Szego, where the right-hand side of (3) has a well-defined form [9], [10], were limited to one-machine problems.

Recent work on the first swing stability of power systems [11]-[12] is concerned with obtaining a practical method for estimating the stability domain without finding the Zubov function. Consider the power system equations with the equilibrium translated to the origin (assume inertias i.e., M_is are absorbed in the nonlinearity)

$$\begin{aligned} \overset{o}{\Theta} &= \tilde{w} \\ \overset{o}{\tilde{w}} &= g(\Theta) \end{aligned} \tag{4}$$

The instability conjecture [11] for the system (4) is that if the projection of the trajectory crosses the surface defined by $\Theta^T g(\Theta) = 0$, then the trajectory is unstable. The time involved in the problem under consideration precludes a first swing analysis however.

APPROXIMATE ZUBOV FUNCTIONS

Recent results in constructive algorithms for determining the stability of a set of matrices [13]-[14] can be used to provide an alternate computational procedure for finding approximate Zubov functions. Proceeding initially as in [13]-[14] write (2) as a "linear" system:

$$\overset{o}{x} = f(x,0) = M(x)x$$

where M(x) is a matrix. If the "linear" system is discretized over the time step h_n (corresponding to the ΔT mentioned previously) to form

$$x_{n+1} = x_n + h_n M(x_n) x_n$$

Let S_M denote the matrices $M(x_n)$ as x_n varies in R^n and consider the set of matrices

$$A \equiv \{I + h_n M_n \mid 0 \le h_n \le h^1,\ M_n \in S_M\}$$

Although the set A is infinite it can be represented by its extreme points. Complex matrices are considered in [13]-[14] but the main results will be summarized here for real matrices.

A set A of nxn real matrices is said to be stable if for every neighborhood of the origin $U \subset R^n$ there exists another neighborhood of the origin V, such that $MV \subseteq U$ for every $M \in A^1$ (the semigroup generated by matrices in A)

Theorem [13]: A set of matrices A is stable if and only if there exists a norm $|\cdot|$, such that for all $M \in A$, and all $x \in R^n$, $|Mx| \le |x|$

Theorem [13]: Given $A = \{M_0, M_1, \ldots, M_{m-1}\}$ is a set of m distinct matrices. Let W_0 be a bounded neighborhood of the origin. Define for $k > 0$

$$W_k = \text{convex hull}\left[\bigcup_{j=0}^{\infty} M_{k}^{j}1,\ W_{k-1}\right]$$

where $k^1 = (k-1) \bmod m$. Then A is stable if and only if $W^* \equiv \bigcup_{i=0}^{\infty} W_i$ is bounded.

W^* is shown to be the norm of the first theorem. A constructive algorithm is given for the computation of W_k using linear programming. In addition conditions are given for stopping the algorithm when either stability or instability is established. The constructive algorithm is begun with some polyhedral W_0. It can be seen [14] that the boundedness of W^* is independent of the choice of W_0.

Several modifications must be made in order to apply the theory to the determination of approximate stability domains. First let the domain of x used to define S_M be limited to

$$L_o = \{x \mid D_o x \le 1\}$$

where $1 \in \mathbb{R}^p$ and let

$$S_M = \{M(x) \mid x \in L_0\}$$

The matrix D_0 must be selected so that at the least every $M \in A$ is stable. If a bounded W^* is produced by the algorithm then it is clear that stability is guaranteed for $x \in \rho W^*$ where ρ is selected as the largest value which yields $\rho W^* \subset L_0$.

The difficulty with the preceeding is that the shape of W^* which is influenced by the selection of W_0 need have no connection with the set L_0. A typical situation is shown in Figure 2.

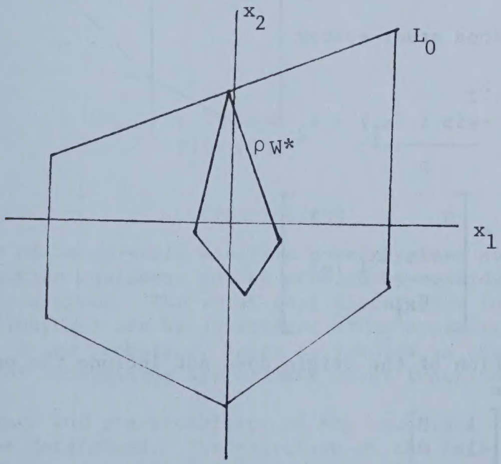


Figure 2

If the region L_0 is regarded as an initial guess for the stability domain (in addition to producing a stable set of matrices A) then it is reasonable to seek a restriction of L_0 with the property that

$$L^* = \{x \mid Mx \in L_0, \text{ for all } M \in A^1\}$$

The largest L^* would then provide an estimate of the stability domain implied by the initial set L_0, Since

$$L^* = \{x \mid D_0 M x \leq 1, \text{ for all } M \in A^1\}$$

a variation on the constructive algorithm can be proposed. If we let

$$D_k = \begin{bmatrix} D_{k-1} \\ D_{k-1} M_1 \\ D_{k-1} M_2 \\ \vdots \\ D_{k-1} M_{m-1} \end{bmatrix}$$

and

$$L_k = \{x \mid D_k x \leq 1\}$$

where $1 \in R^{mp}$ then the sequence L_k defines a set of regions that contract (as the W_k expand)

The modification is intimately connected with the original algorithm in that the original algorithm works with the convex hull of points defining the regions W_k while the suggested modification alters the system of linear inequalities whose vertices are the points. It seems likely that the finite termination property of the original algorithm will carry over to the modification. It is also clear that there are large numbers of ineffective constraints in the expression for L_k which can be eliminated. A simple example indicates the degree of approximation involved.

EXAMPLE

Consider the second order system

$$\overset{\circ}{x}_1 = x_2$$
$$\overset{\circ}{x}_2 = \frac{-\sin(\pi x_1)}{\pi} - x_2$$

where

$$M(x) = \begin{bmatrix} 0 & 1 \\ \dfrac{-\sin \pi x_1}{\pi x_1} & -1 \end{bmatrix}$$

The domain of attraction of the origin does not include the points $x_1 = \pm 1$. If we take

$$D_0 = \begin{bmatrix} 1 & 0 \\ -1 & 0 \end{bmatrix}$$

the external matrices of A are

$$M_1 = I + h\begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$$

$$M_2 = I + h\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}$$

Using a coarse value for h for simplicity (h=.5)

$$M_1 = \begin{bmatrix} 1 & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

$$M_2 = \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$$

The fact that M_2 has an eigen value on the unit circle and that

$$M_2{}^k = \begin{bmatrix} 1 & 1-\frac{1}{2}^k \\ 0 & \frac{1}{2}^k \end{bmatrix}$$

causes the lines in Figure 3 to move asympotoically to the boundarys. Including the limiting matrix M_2^∞ in the set gives the domain L* as shown.

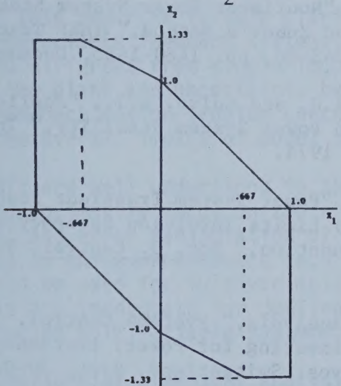


Figure 3

CONCLUSIONS

Stability of large-scale electric power systems subject to mis-operation of protection equipment can be studied by considering a form of uncertain dynamic systems. The worst case disturbance (sequence of line openings and closings) can be determined if an appropriate Zubov function can be found. A new technique, based on results in the stability of sets of matrices, for determining approximate Zubov functions has been suggested.

The accuracy and practicability of the technique for large scale systems must be determined. The selection of the initial estimate for the region and techniques to eliminating the non-effective constraints must also be investigated.

REFERENCES

[1] Phadke, A.G. "Recent Development in Digital Computer Based Protection and Control in Electric Power Susbstations", presented at Conference on Power System Protection, The Institution of Engineers Madras, India, April 1980.

[2] Boyle, J.R., Edwards, H.A. and Khalifa, S.A. "TVA's Present Reclosing Practices", presented at The 1979 Protective Relaying Conference, Georgia Institute of Technology, May 1979.

[3] Gutman, S., "Uncertain Dynamical Systems, A Lyapunov Min-Max Approach", IEEE Trans. on Auto. Control, AC-24, No. 3, 1979.

[4] Zubov, V.I., Mathematical Methods for the Study of Automatic Control Systems, New York, Pergamon Press, 1963.

[5] Hahn, W., Theory and Application of Lyapunov's Direct Method, Englewood Cliffs, N.J., Prentice-Hall, 1963.

[6] Margolis, S.G. and Vogt, W.G., "Control Engineering Applications of V.I. Zubov's Construction Procedure for Lyapunov functions," IEEE Trans. Automatic Control, Vol. AC-8, pp. 104-113, April, 1963.

[7] Kerr,C.N., discussion of [14], IEEE Trans. Automatic Control, Vol. AC-9, p. 196, April, 1964.

[8] Yu, Y.N. and Vongsuriya, K., "Nonlinear Power System Stability Study by Lyapunov Function and Zubov's Method," IEEE Trans, Power Apparatus and Systems, Vol. PAS-86, pp. 1480-1485, December, 1967.

[9] Prabhakara, F.S., El-Abiad, A.H. and Koivo, A.J., "Application of Generalized Zubov's Method to Power System Stability," Int. J.Control, Vol. 20, No. 2, pp. 203-212, 1974.

[10] Prusty, S. and Sharma, G.D., "Power System Transient Stability Regions: Transient Stability Limits Involving Saliency and the Optimized Szego's Lyapunov Function," Int. J. Control, Vol. 19, No. 2, pp. 373-384, 1974.

[11] "Transient Energy Stability Analysis," Systems Control, Inc., presented at the Systems Engineering for Power: Emergency Operating State Control Conference, Davos, Switzerland, Sept. 30-Oct.5, 1979. publication CONF-790904--P1 (available from NTIS)

[12] Kakimoto, N, Ohsawa, Y. and Hayashi, M. "Transient Stability Analysis of Electric Power System via Lurè Type Lyapunov Function", Trans. IEE of Japan Vol, 98, No. 516, May/June, 1978.

[13] Brayton, R.K. and Tong, C.H., "Stability of Dynamical Systems: A Constructive Approach," IEEE Trans. on Circuits and Systems, April 1979, 224-234.

[14] Brayton, R.K. and Tong, C.H., "Constructive Stability and Asymptotic Stability of Dynamical Systems", IBM Research Report RC 7909, Oct. 1979.

TRADEOFFS IN THE DESIGN OF MULTIVARIABLE FEEDBACK SYSTEMS*

J.E. Wall, Jr.
Honeywell SRC
Minneapolis, MN

J.C. DOYLE, Consultant
Honeywell SRC
Minneapolis, MN, and
University of California
Berkeley, CA

C.A. Harvey
Honeywell SRC
Minneapolis, MN

ABSTRACT

Some of the tradeoffs involved in the design of multivariable feedback systems are examined. The paper addresses the problem of obtaining good loop properties at both the plant input and output and the impact of non-minimum phase zeros.

Notation:

$\bar{\sigma}(\cdot)$ - maximum singular value
$\underline{\sigma}(\cdot)$ - minimum singular value
$\bar{\lambda}(\cdot)$ - eigenvalue
$(\cdot)^T$ - transpose
$\overline{(\cdot)}$ - complex conjugate
$(\cdot)^*$ - complex conjugate transpose
diag $[\cdot]$ - diagonal matrix

I. INTRODUCTION

The intent of this paper is to help clarify the tradeoffs involved in the design of multivariable feedback systems. In one respect, multivariable systems are no different from single-loop systems. There is always uncertainty in the plant and uncertainty between design models and the actual plant. The feedback design problem centers around the tradeoffs involved in reducing the overall impact of model uncertainty on system performance.

These tradeoffs are well understood in the single-loop case, and classical control methods primarily focus on their analysis. Actual classical design involves trial and error manipulation of compensators using frequency-domain analysis tools to guide and evaluate each trial design. A similar design procedure could be used for multivariable systems, except that the classical analysis tools are inadequate for dealing with the fundamental feedback properties of coupled multivariable systems.

Some generalizations of classical analysis methods to the multivariable are found in [1]. Specifically, the algebraic tradeoff between large loop gain for good sensitivity and small loop gain for stability margins is discussed. The tradeoffs imposed on multivariable systems by the Bode gain/phase relations are also examined in [1]. These results are briefly reviewed in Section II. Sections III and IV discuss other multivariable phenomena in more detail. The intrinsically multivariable problem of obtaining loop properties (e.g., stability margins) at different points in a multivariable feedback configuration is considered in Section III. Section IV examines the effect of non-minimum phase zeros on multivariable systems. Some of these ideas are illustrated through an example of an electric generating plant in Section V. The conclusions are in Section VI.

---

* This research was supported by DOE Contract ET-78-C-01-3391.

II. MULTIVARIABLE GENERALIZATIONS OF SINGLE LOOP PROPERTIES

Consider the feedback configuration shown in Figure 1. This represents a (nominally) stable closed-loop system consisting of a linear time-invariant plant $G(s)$ and controller $K(s)$. In this section, we will discuss multivariable generalizations of some classical properties of single-loop feedback systems.

One use of feedback is to reduce sensitivity to changes in the plant. It is well-known that one useful sensitivity function for a single-loop feedback system is the inverse of the return difference. To quantify the desensitizing effect of feedback in the multivariable case, we use the comparison sensitivity approach [2]. For the comparison, let G represent the nominal plant and $G(I+\Delta)^{-1}$ represent the perturbed plant. To obtain a direct comparison of sensitivity with and without feedback, we consider the feedback system with precompensation $(I+KG)$ shown in Figure 2. This precompensation is added so that the closed-loop transfer function for the nominal plant is identical with the nominal open-loop transfer function, G. The perturbed closed-loop transfer function is

$$\begin{aligned} G(I+\Delta)^{-1}(I+KG(I+\Delta))^{-1}(I+KG) &= G(I+(I+KG)^{-1}\Delta)^{-1} \\ &= G(I+\Delta_{CL})^{-1} \end{aligned} \qquad (2.1)$$

where the effective closed-loop perturbations Δ_{CL} is

$$\Delta_{CL} = (I+KG)^{-1}\Delta \qquad (2.2)$$

Thus for multivariable systems, the inverse return difference matrix, $(I+KG)^{-1}$, relates the sensitivity of the closed-loop system to that of the open-loop system. In terms of the spectral norm, if the condition

$$1 > \bar{\sigma}[(I+KG)^{-1}] \qquad (2.3)$$

or, equivalently,

$$\underline{\sigma}(I+KG)>1 \qquad (2.4)$$

holds at some frequency, then the use of feedback accomplishes desensitization at that frequency.

Feedback can also reduce the effect of a disturbance, d, on the output, y (Figure 1). The response at the output to a disturbance at the plant input is

$$y = G(I+KG)^{-1}d \qquad (2.5)$$

Since the open-loop response to such a disturbance is simply Gd, satisfaction of condition (2.4) at some frequencies means that feedback provides some disturbance rejection at those frequencies.

In terms of sensitivity and disturbance rejection, good performance of a multivariable (as well as single-loop) feedback system translates into the requirement of a "large" return difference matrix. Since

$$|\underline{\sigma}(I+KG) - \underline{\sigma}(KG)|<1 \qquad (2.6)$$

this is equivalent to requiring "large" loop gain. A more detailed discussion of the relationship between performance and loop gain may be found in [1]. We will return to the desideratum of large loop gain after considering stability margins for multivariable feedback systems.

The representation of uncertainty as $(I+\Delta)^{-1}$ is convenient for studying sensitivity. Some uncertainties in physical systems, however, cannot be represented in this way. In particular, both gain and phase become highly uncertain at high frequencies. Modeling the plant with uncertainty as $G(I+\Delta)$ is quite useful in these situations.

Using this representation for the plant with uncertainty in the multivariable feedback system of Figure 1, it has been shown [3] that the closed-loop system remains stable for all (stable) perturbations $\Delta(s)$ satisfying

$$\bar{\sigma}(\Delta(jw)) < \underline{\sigma}[I+(K(jw)G(jw))^{-1}] \quad \forall\ w \geq o \qquad (2.7)$$

This condition can be obtained as a consequence of the Small Gain Theorem [4] or the multivariable Nyquist criterion [5]. The size of $I+(KG)^{-1}$ measures the robustness of closed-loop stability with respect to multiplicative perturbations $(I+\Delta)$. This is directly analogous to the distance to the critical point on an inverse Nyquist diagram, $\left|1+\frac{1}{KG}\right|$.

Adequate representations of physical systems inevitably require that the perturbation become large at high frequency, and so large $(KG)^{-1}$ (or, equivalently, small loop gain KG) is required at these frequencies to ensure stability.

This last statement leads to a trade-off in the design of feedback systems. At a particular frequency, it is possible to have large loop gain for desensitization and disturbance rejection or small loop gain for stability margins -- but not both. More precisely, consider the identity

$$(I+KG)^{-1} + (I+(KG)^{-1})^{-1} = I \qquad (2.8)$$

A direct consequence of this identity is the string of inequalities

$$\frac{\underline{\sigma}(I+L)}{\underline{\sigma}(I+L)+1} \leq \underline{\sigma}(I+L^{-1}) \leq \frac{\underline{\sigma}(I+L)}{\underline{\sigma}(I+L)-1} \qquad (2.9)$$

where either KG or $(KG)^{-1}$ may be substituted for L, and the right-hand inequality requires $\underline{\sigma}(I+L) > 1$. Clearly then,

$$\underline{\sigma}(I+L) >> 1 \longrightarrow \underline{\sigma}(I+L^{-1}) \simeq 1$$

Thus, desensitization can be obtained only at frequencies where stability margins need not be large. Since large stability margins are always required at sufficiently high frequencies, no desensitization can be achieved there.

As an aside, note that the left-hand inequality in (2.9) may be rewritten as

$$\underline{\sigma}(I+L) \leq \frac{\underline{\sigma}(I+L^{-1})}{1-\underline{\sigma}(I+L^{-1})} \qquad (2.10)$$

for $\underline{\sigma}(I+L^{-1})<1$. Hence

$$\underline{\sigma}(I+L^{-1}) << 1 \longrightarrow \underline{\sigma}(I+L) << 1$$

Recall that L may be replaced by either KG or $(KG)^{-1}$. So roughly speaking, "poor" sensitivity at any frequency is equivalent to "poor" margins there.

We have seen that the trade-off between desensitization and stability margins applies to multivariable feedback systems as well as to single-loop systems. Typically, desensitization and disturbance rejection are desired at low frequencies, and large stability margins are required at high frequencies. A critical factor in the design process is the necessity to make the transition from large loop gain to small loop gain. Throughout this transition region, poor sensitivity and poor stability margins must be avoided.

The sensitivity and stability margins of a single-loop feedback system near crossover are known to be highly dependent on the phase of the loop transfer function. Also, Bode has shown that for a class of functions which includes proper, rational functions with no poles or zeros in the right-half plane, either the gain or phase is sufficient to reconstruct the entire function [6]. Thus the necessity to avoid poor sensitivity and poor stability margins near crossover of single-loop feedback systems (i.e., carefully control the phase of KG) imposes limitations on $|KG|$ (and, in turn, on $|1+KG|$ and $|1+\frac{1}{KG}|$) away from crossover. The Bode gain/phase relation provides a fundamental limitation on the achievable performance of single-loop feedback systems.

The multivariable generalization of the gain/phase relation is complicated by the fact that the singular values of a rational transfer function matrix are not analytic. The eigenvalues of such a matrix, however, do satisfy a gain/phase relation and can be related to the singular values by the inequalities [1].

$$\underline{\sigma}(I+L) \leq |\lambda_i(I+L)| \leq \overline{\sigma}(I+L) \quad \forall i \tag{2.11}$$

The resulting restrictions on the singular values are just as severe as in the single-loop case. The conclusion is that multivariable systems have limitations on their achievable performance analogous to those imposed on single-loop systems by the Bode gain/phase relations.

Summarizing this section, classical sensitivity and stability margins for single-loop systems have natural multivariable generalizations. The tradeoff between sensitivity and stability margins at any frequency is reflected in the algebraic constraint (2.8). The Bode gain/phase relation also has a natural multivariable generalization and introduces tradeoffs across frequencies. These multivariable generalizations, of course, come with additional complicating factors compared to the single-loop case. However, the fundamental classical tradeoffs involved in the design of feedback systems naturally extend to the multivariable case.

III. DIRECTIONALITY IN MULTIVARIABLE FEEDBACK SYSTEMS

The preceding section discussed frequency-domain tradeoffs which occur in the design of single-loop and multivariable feedback systems. In this section, we examine some tradeoffs that are unique to multivariable systems.

For multivariable systems, unlike scalar systems, signals have a "spatial" as well as frequency distribution. Some sensors are noisier than others. Certain actuators might saturate at lower signal levels than others. Disturbances enter particular loops, but not others, etc. One is lead to consider, for example, different "bandwidths" in various loops.

The stability condition (2.7) is tight if all that is known about the perturbation Δ is its maximum singular value. However, if more is known about Δ — large in one direction, small in another — then condition (2.7) can be sharpened through the use of frequency-dependent weighting matrices [7]. That is, for a perturbed plant $G(s)(I+\Delta(s))$, the closed-loop system remains stable for all (stable) perturbations $\Delta(s)$ satisfying

$$\overline{\sigma}[Q_I(jw)\Delta(jw)R_I^{-1}(jw)] < \underline{\sigma}[R_I(jw)(I+(K(jw)G(jw)^{-1})Q_I^{-1}(jw)] \quad \forall w \geq o \tag{3.1}$$

The known characteristics of the allowable perturbations are reflected in the (nonsingular) weighting matrices R_I and Q_I.

The stability margin condition (3.1) yields margins at a particular point in the feedback configuration of Figure 1. These margins are at the inputs to the plant. The weighting matrices in (3.1) are useful in handling the issue of directionality at this point. For single-loop systems, margins at the input are identically equivalent to margins at the output. Input and output margins are not equivalent, however, in the multivariable case. Output margins are obtained by considering a perturbed plant $(I+\Delta_o(s))G(s)$. The stability condition (using weighting matrices R_o and Q_o) may be written as

$$\bar{\sigma}[Q_o(jw)\Delta_o(jw)R_o^{-1}(jw)] < \underline{\sigma}[R_o(jw)(I+(G(jw)K(jw))^{-1})Q_o^{-1}(jw)] \ \forall w \geq o \quad (3.2)$$

This is the same condition as (3.1), with G and K interchanged.

From a design viewpoint, good margins are typically desired at both the inputs and the outputs of the plant. One may express this objective slightly differently by first fixing the input margins and then examining the limits imposed on the output margins. More precisely, we pose the following formal optimization problem at a single frequency:

$$\begin{aligned} &\max_{K} \ \underline{\sigma}[R_o(I+(GK)^{-1})Q_o^{-1}] \\ &\text{s.t. } \sigma_i[R_I(I+(KG)^{-1})Q_I^{-1}] = 1 \quad \forall i \end{aligned} \qquad (P)$$

and now we require both G and K to be square.

This constraint of all singular values equal to 1 is equivalent to constraining the matrix to be unitary. Hence

$$I+(KG)^{-1} = R_I^{-1}UQ_I \qquad (3.3)$$

for some unitary U. The function to be maximized can then be written as

$$\begin{aligned} \underline{\sigma}[R_o(I+(GK)^{-1})Q_o^{-1}] &= \underline{\sigma}[R_oG(I+(KG)^{-1})G^{-1}Q_o^{-1}] \qquad (3.4) \\ &= \underline{\sigma}[(R_oGR_I^{-1})U(Q_oGQ_I^{-1})^{-1}] \end{aligned}$$

This means the original problem (P) is equivalent to

$$\begin{aligned} &\max_{U} \ \underline{\sigma}[(R_oGR_I^{-1})U(Q_oGQ_I^{-1})^{-1}] \\ &\text{s.t. } UU^* = I \end{aligned} \qquad (P^1)$$

To solve (P^1), let $R_oGR_I^{-1} = U_1\Sigma_1V_1^*$ and $Q_oGQ_I^{-1} = U_2\Sigma_2V_2^*$ be singular value decompositions where

$$\Sigma_1 = \text{diag}[\alpha_1,\alpha_2,\ldots,\alpha_n] \quad , \quad \alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$$

$$\Sigma_2 = \text{diag}[\beta_1,\beta_2,\ldots,\beta_n] \quad , \quad \beta_1 \geq \beta_2 \geq \cdots \geq \beta_n$$

Making these substitutions into (P^1),

$$\underline{\sigma}[(R_oGR_I^{-1})U(Q_oGQ_I^{-1})^{-1}] = \underline{\sigma}[\Sigma_1\hat{U}\Sigma_2^{-1}] \qquad (3.5)$$

where $\hat{U} = V_1^*UV_2$ is a unitary matrix.

Proposition

$$\begin{aligned} &\max_{\hat{U}} \ \underline{\sigma}[\Sigma_1\hat{U}\Sigma_2^{-1}] = \min \left(\frac{\alpha_i}{\beta_i}\right) \\ &\text{s.t. } \hat{U}\hat{U}^* = I \end{aligned}$$

The formal problem (P) can be interpreted in terms of this proposition. The best possible output margins equal the minimum of the ordered ratio of the singular values of $R_O G R_I^{-1}$ to the singular values of $Q_O G Q_I^{-1}$. When the uncertainty at the input and output is unstructured, i.e., R_I, R_O, Q_I, Q_O are all scalar multiples of the identity, the maximum achievable margins at the outputs equal 1 — the same as the margins at the inputs. This unstructured case causes the multivariable problem to be analogous to the single-loop case. When the uncertainties do have some directionality or structure (as reflected in non-identity R's and Q's), the feedback problem is inherently multivariable. This directionality may impose additional limitations on the achievable performance.

In the Appendix, it is shown that using $\hat{U}$ equal to the identity in the proposition achieves the maximum. With this choice of $\hat{U}$, we can solve for K from (3.3) as

$$K = (R_I^{-1} V_1 V_2^* Q_I - I)^{-1} G^{-1} \tag{3.6}$$

assuming that $\hat{U} = I$ yields $R_I^{-1} V_1 V_2^* Q_I - I$ invertible. As far as actually achieving this limit at each frequency, note that V_1 and V_2^* need not be rational functions of frequency. Hence, in general, we cannot achieve this limit with a causal compensator K(s). The question of the limitations imposed on the output margins by both fixed input margins and a causal compensator is still open.

In concluding this section, we point out that the formal problem considered here is not especially important in and of itself. One could pose several other such problems, e.g. maximizing $\underline{\sigma}(I+KG)$ subject to constraints on I+GK, etc. The reason for posing this problem is that it illustrates how directionality can introduce uniquely multivariable phenomena.

IV. NON-MINIMUM PHASE BEHAVIOR IN MULTIVARIABLE SYSTEMS

Given a desired rational, strictly proper loop transfer function H(s) that yields a stable closed-loop system, there exists rational, strictly proper compensation K(s) for the minimum phase plant G(s) such that the desired loop transfer function is fit arbitrarily closely and the resulting closed-loop system is stable. Indeed, it suffices to introduce the feedback compensation

$$K(s) = G^{-1}(s) H(s) \left(\frac{\alpha}{s+\alpha}\right)^m \tag{4.1}$$

where m is chosen to make K(s) strictly proper and α is chosen sufficiently large. This means that if H(s) is designed to achieve the sensitivity and stability margin tradeoff discussed in Section II, then any minimum phase plant can be compensated to realize this loop transfer function as closely as desired. The same result applies to multivariable systems as well as single-loop systems by interpreting (4.1) in a matrix sense. In the multivariable case, however, such compensation yields the desired loop properties only at the plant inputs; the output loop properties can be quite different.

This is not true for non-minimum phase systems. A transfer function with right-half-plane zeros may be factored as

$$G(s) = M(s) P(s) \tag{4.2}$$

where M(s) is minimum phase and P(s) is an all-pass ($|P(jw)| = 1, \forall w$). The all-pass cannot be cancelled as in (4.1). It contributes a negative phase angle and thereby aggravates the tradeoff problem. Roughly speaking, a non-minimum phase zero λ requires small loop gain throughout some frequency interval "near" λ[8]. That is, the advantages of feedback are achievable at frequencies below and above the non-minimum phase zero's magnitude but not around it.

Multivariable non-minimum phase systems are subject to analogous restrictions. A plant transfer matrix with right-half-plane zeros may be factored in two ways as

$$G(s) = M_1(s)\, P_1(s) = P_2(s) M_2(s) \tag{4.3}$$

where the P's are all-pass ($P^*(jw)P(jw) = I \ \forall w$) and the M's are minimum phase. These factorizations can be expressed in state-space terms. Let G(s) be the transfer matrix of the quadruple (A,B,C,D). Then for our purposes, G(s) has a zero at λ means

$$\begin{bmatrix} \lambda I - A & -B \\ -C & -D \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} = 0 \tag{4.4}$$

has a solution with $w^*w = 1$. For a single real zero at λ, one factorization (unique up to a unitary transformation) $M_1(s)P_1(s)$ is given by

$$P_1(s) = I - \left(\frac{2\lambda}{s+\lambda}\right) ww^T \tag{4.5}$$

and $M_1(s)$ is the transfer matrix of the quadruple $(A,\hat{B},C,D)$ where

$$\hat{B} = B - 2\lambda x w^T \tag{4.6}$$

For the case of a single pair of complex zeros at λ and λ^*,

$$P_1(s) = I - \left(\frac{4\alpha\gamma}{s^2+2\alpha s + \lambda\lambda^*}\right)\left[s\,\mathrm{Re}\left(ww^* - \frac{\alpha\beta}{\lambda}\bar{w}w^*\right) + \mathrm{Re}\left(\lambda ww^* - \alpha\beta\bar{w}w^*\right)\right] \tag{4.7}$$

$$\hat{B} = B - 2\alpha\gamma\left[xw^* - \frac{\alpha\beta}{\lambda}\bar{x}w^*)\right] \tag{4.8}$$

where $\alpha = \mathrm{Re}(\lambda)$, $\beta = w^Tw$, and $\gamma = \left(1 - \frac{\alpha^2\beta^2}{\lambda\lambda^*}\right)^{-1}$. The proofs of (4.5) - (4.8) follow from the generalized eigenvalue problem (4.4) with algebraic manipulation. Multiple non-minimum phase zeros can be handled by repeated application of (4.5) - (4.8). We note that similar formulas may be obtained for the factorization $P_2(s)M_2(s)$ by using the left generalized eigenvector.

$$\begin{bmatrix} \hat{x}^T & v^T \end{bmatrix} \begin{bmatrix} \lambda I - A & -B \\ -C & -D \end{bmatrix} = 0 \tag{4.9}$$

The non-minimum phase zero λ adds extra phase lag in the input direction w and the output direction v. This is the same effect and has the same restrictions for feedback design as in the single-loop case. However, the transfer matrix from inputs orthogonal to w to outputs orthogonal to v does not have any additional phase lag caused by λ. Thus, the non-minimum zero λ does not impose any limitations on the loop gain in these orthogonal directions.

Because a non-minimum phase zero λ does not impose limitations in directions orthogonal to w and v, it could be the case that a feedback design has large loop gain in these directions*. Such a design would necessarily have a large spread in the singular values of KG at frequencies near λ. This large spread is the same situation that occurs in the presence of structured uncertainty, i.e., weighting matrices R_I and Q_I not equal to a scalar multiple of the identity. As discussed in Section III, the tradeoff between loop properties at the plant input vs. the output enters the design problem when there is a spread in the singular values of KG. This additional tradeoff is uniquely multivariable.

---

* Other considerations may restrict this loop gain. For example, the transfer matrix associated with this subspace may itself be non-minimum phase, although not having a zero at λ.

Thus, non-minimum phase zeros affect multivariable feedback design in two ways. First, they contribute additional phase lag in certain input and output directions. This limits the achievable performance in these directions and is directly analogous to non-minimum phase zeros in single-loop systems. The resulting unequal gains in different directions can lead to a conflict between input and output loop properties. This second effect has no single-loop analog.

V. AN ELECTRIC GENERATING PLANT EXAMPLE

The impact of a non-minimum phase zero on multivariable feedback design will be illustrated in this section by comparing two feedback designs for an electric generating plant. The plant is Philadelphia Electric Company's Cromby Number 2 unit, a 250 MW coal-fired boiler-turbine-generator system. For design pruposes, we use a linearized version of the nonlinear 14th order model of this unit described in [9], [10]. The model is numerically linearized at steady-state conditions and a generation of 220 MW. The six control variables and six output variables used in the design are listed in Table 1. In order to regulate the output variables, six integrators are appended to the original model. This model has a non-minimum phase zero at 0.35 rad/sec. The non-minimum phase input vector w has non-zero components in the air flow and both burner tilts.

Table 1. Cromby Number 2 Control and Output Variables Used For Controller Design

Controls:	Outputs:
Length of the coal feeder stroke	Generation
Governor valve	Drum level
Feedwater valve	Secondary superheater pressure
Air flow	Secondary superheater temperature
Superheat burner tilt	Reheat temperature
Reheat burner tilt	Coal flow

The design approach used here is a two-step LQG procedure. The first step is the design of an acceptable LQ state-feedback regulator. The second step consists of employing the asymptotic filter design procedure described in [11]. This filter design procedure asymptotically yields full-state loop properties at the inputs for a minimum phase plant. We will now illustrate what can happen with a non-minimum phase zero.

Figure 3 and 4 show the singular values of the return difference matrix with the loops broken at the plant inputs under state-feedback for two different LQ designs. Since these are LQ designs, the minimum singular value is guaranteed to be no less than one. The design in Figure 3 does not roll off the non-minimum phase direction before 0.35 rad/sec; the design in Figure 4 does provide this roll off. The second design has rolled off both burner tilts and the air flow by about 0.1 rad/sec.

The results of using the asymptotic filter designs with these two state feedback gains are shown in Figures 5 and 6. These figures show the singular values of the return difference matrix with the loops broken at the inputs under output-feedback. The impact of the non-minimum phase zero is evident. Figure 5 shows a minimum singular value less than one from 0.1 rad/sec to almost 10 rad/sec. This feedback design amplifies disturbances (in some direction) and increases sensitivity in this frequency range.

From Figure 6, the design which rolled off the non-minimum phase direction does not have this problem. The asymptotic filter design procedure was able to achieve the full-state loop properties in this case.

These results are presented for illustrative purposes only. The controllers are not intended to be considered as a practical design. They are intended only to provide information on the impact of non-minimum phase zeros.

VI. CONCLUSIONS

This paper has discussed design tradeoffs which impact the achievable performance of multivariable feedback systems. Multivariable generalizations of the well-understood single-loop sensitivity vs. stability margins tradeoff and Bode gain/phase relations were reviewed in Section II. Using these "singular value" characterizations of loop properties, the intrinsically multivariable problem of obtaining good loop properties at both the plant inputs and outputs was examined. This tradeoff was illustrated by considering the question of finding the best output margins for given input margins. In Section IV, the influence of non-minimum phase zeros was discussed. These zeros add extra phase lag, just as in the single-loop case, and also may lead to a conflict between desired loop properties at different points in the feedback configuration. In closing, we point out that the approach taken here allows a great deal of single-loop intuition to be carried over to the multivariable case. What is less satisfying, is that the results presented are fragmented; essentially one-at-a-time results. Much work remains to be done to unify these results in a coherent theory.

REFERENCES

1. Doyle, J.C. and Stein, G., "Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis," to appear IEEE Trans. Automat. Contr., February, 1981

2. Cruz, J.B. and Perkins, W.R., "A New Approach to the Sensitivity Problem in Multivariable Feedback System Design," IEEE Trans. Automat. Contr., vol. AC-9, pp 216-223, 1964.

3. Doyle, J.C., "Robustness of Multiloop Linear Feedback Systems," Proc. 1978 CDC, San Diego, Calif., January, 1978.

4. Desoer, C.A. and Vidyasagar, M., Feedback Systems: Input-Output Properties, Academic Press, N.Y., (1975).

5. Rosenbrock, H.H., "The Stability of Multivariable Systems," IEEE Trans. Automat. Contr., vol. AC-17, pp. 105-107, 1972.

6. Bode, H.W., Network Analysis and Feedback Amplifier Design, D. Van Nostrand, Princeton, 1945.

7. Safonov, M.G., "Tight Bounds on the Response of Multivariable Systems with Component Uncertainty," 16th Allerton Conference, October, 1978.

8. Horowitz, I.M., Synthesis of Feedback Systems, Academic Press, N.Y., (1963).

9. McDonald, J.P., Kwatny, H.G., and Spare, J.H., "A Nonlinear Model for Reheat Boiler-Turbine-Generator Systems, Part I - General Description and Evaluation," Proc. 12th JACC, pp. 219-226, 1971.

10. Kwatny, H.G., McDonald, J.P., and Spare, J.H., "A Nonlinear Model for Reheat Boiler-Turbine-Generator Systems, Part II - Development," Proc. 12th JACC, pp. 227-236, 1971.

11. Doyle, J.C. and Stein, G., "Robustness with Observers," IEEE Trans. Automat. Contr., vol. AC-24, pp. 607-611, 1979

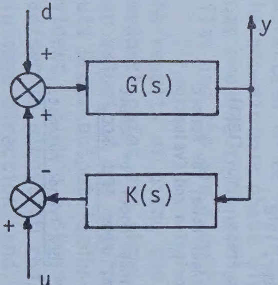


Figure 1
Multivariable
Feedback System

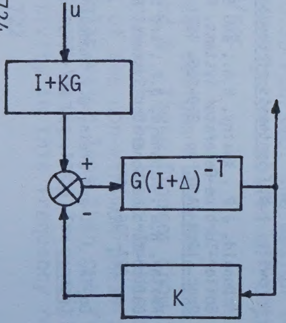


Figure 2
Precompensation For
Comparative Sensitivity

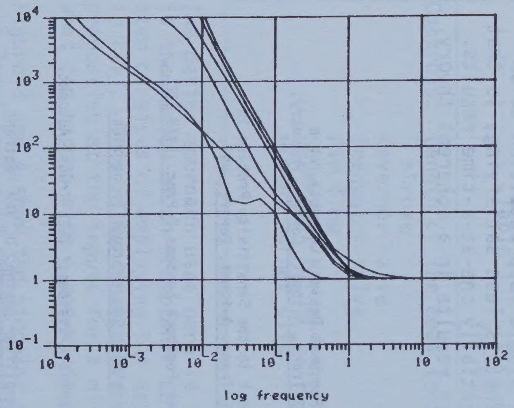


Figure 3 Singular values of the return difference matrix under state-feedback for desinn 1.

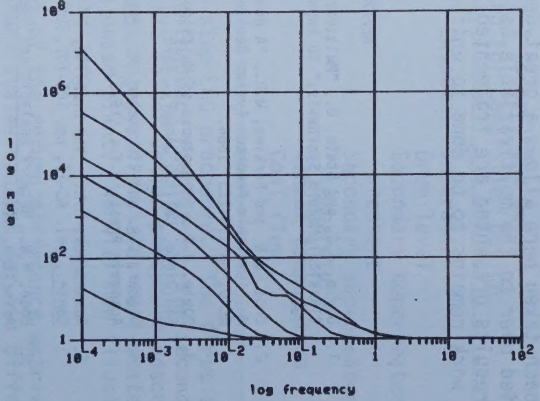


Figure 4 Singular values of the return difference matrix under state-feedback for design 2.

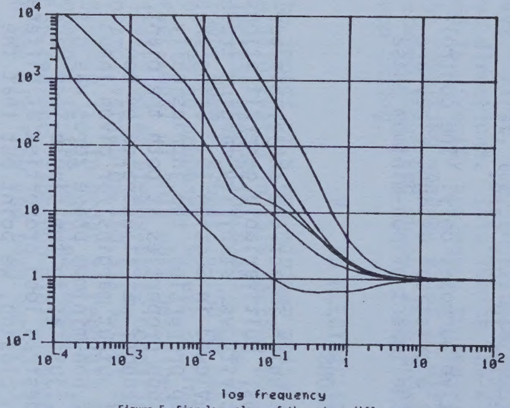


Figure 5 Singular values of the return difference matrix under output-feedback for desinn 1.

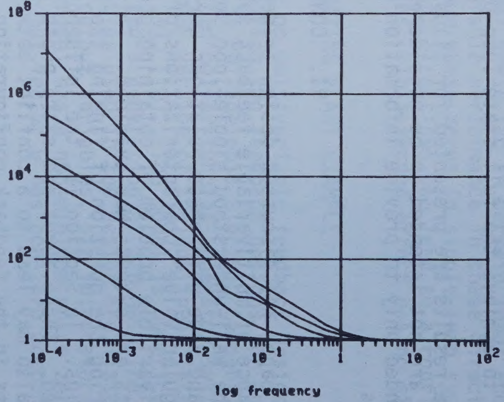


Figure 6 Singular values of the return difference matrix under output-feedback for design 2.

APPENDIX

Proposition: Let $\Sigma_1 = \text{dia}[\alpha_1,\alpha_2,\ldots,\alpha_n]$, $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ and $\Sigma_2 = \text{diag}[\beta_1,\beta_2,\ldots,\beta_n]$, $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$. Then

$$\max_{\hat{U}} \underline{\sigma}[\Sigma_1 U \Sigma_2^{-1}] = \min_{i} \left(\frac{\alpha i}{\beta_i}\right)$$

$$\text{s.t. } \hat{U}\hat{U}^* = I$$

Proof: For any unitary $\hat{U}$,

$$\overline{\sigma}[\Sigma_2 \hat{U}^* \Sigma_1^{-1}] = |\Sigma_2 \hat{U}^* \Sigma_1^{-1}| \tag{A.1}$$

$$\geq \frac{\beta_i}{\alpha_i} \cdot |\text{diag}[\underbrace{1,\ldots,1}_{\text{i times}},0,\ldots,0]\ \hat{U}^* \text{diag}[0,\ldots,0,\underbrace{1,\ldots,1}_{\text{n-i+1 times}}]|$$

where $\frac{\beta i}{\alpha_i} = \max_{j} ({}^{\beta}j/\alpha_j)$. By compatibly partitioning $\hat{U}$, this may be written as

$$\overline{\sigma}[\Sigma_2 \hat{U}^* \Sigma_1^{-1}] \geq \frac{\beta_i}{\alpha_i} \cdot \left| \begin{pmatrix} I & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{U}_1^* & \hat{U}_4^* & \hat{U}_7^* \\ \hat{U}_2^* & \hat{U}_5^* & \hat{U}_8^* \\ \hat{U}_3^* & \hat{U}_6^* & \hat{U}_9^* \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix} \right|$$

$$= \frac{\beta_i}{\alpha_i} \cdot \left| \begin{pmatrix} 0 & \hat{U}_4^* & \hat{U}_7^* \\ 0 & \hat{U}_5^* & \hat{U}_8^* \\ 0 & 0 & 0 \end{pmatrix} \right| \tag{A.2}$$

But

$$\begin{pmatrix} 0 & 0 & 0 \\ \hat{U}_4 & \hat{U}_5 & 0 \\ \hat{U}_7 & \hat{U}_8 & 0 \end{pmatrix} \begin{pmatrix} 0 & \hat{U}_4^* & \hat{U}_7^* \\ 0 & \hat{U}_5^* & \hat{U}_8^* \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1-\hat{U}_6\hat{U}_6^* & -\hat{U}_6\hat{U}_9^* \\ 0 & -\hat{U}_9\hat{U}_6 & I-\hat{U}_9\hat{U}_9 \end{pmatrix}$$

$$= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & I \end{pmatrix} - \begin{pmatrix} 0 \\ \hat{U}_6 \\ \hat{U}_9 \end{pmatrix} \begin{pmatrix} 0 & \hat{U}_6^* & \hat{U}_9^* \end{pmatrix} \tag{A.3}$$

Hence the norm of the matrix on the rhs of (A.2) is one, and so

$$\overline{\sigma}[\Sigma_2 \hat{U}^* \Sigma_1^{-1}] \geq \frac{\beta_i}{\alpha_i} \tag{A.4}$$

Equation (A.4) is true for all unitary $\hat{U}^*$, with equality being obtained for $\hat{U}^* = I$. So

$$\max_{\hat{U}} \underline{\sigma}[\Sigma_1 \hat{U} \Sigma_2^{-1}] = \{\min_{\hat{U}} \overline{\sigma}[\Sigma_2 U^* \Sigma_1^{-1}]\}^{-1} = \min_{i} \left(\frac{\alpha_i}{\beta_i}\right)$$

Q.E.D.

GENERALIZED DECOMPOSITIONS AND STABILITY OF NONLINEAR SYSTEMS*

M. IKEDA
Kobe University
Kobe 657 Japan

D. D. ŠILJAK
University of Santa Clara
Santa Clara, California 95053

ABSTRACT

The Inclusion Principle for dynamic systems is extended to consider the expansions and contractions of Lur'e-Postnikov systems. The obtained results are used to reinforce the generalized decomposition schemes used in stability studies of multimachine power systems via vector Liapunov functions.

1. INTRODUCTION

A motivation for this work has been provided by the applications of decomposition-aggregation schemes to transient stability studies of multimachine power systems. A distinct feature of the decomposition schemes has been the presence of the reference machine in each group of machines representing a subsystem. For this reason, the standard vector Liapunov functions could not be constructed without expanding the state space of the overlapping subsystems into a larger space in which the subsystems appear as disjoint. A calculation of stability regions in the expanded space could be carried out by the standard techniques and then interpreted in the original space. The purpose of this paper is to present a deeper explanation of this procedure using the Principle of Inclusion.

The Inclusion Principle has been formulated for the expansions and contractions of dynamic systems to provide conditions under which the behavior of a system of smaller dimension can be extracted from a higher dimensional system and, alternatively, under which a small system can represent the corresponding system of higher dimension. The Principle has been applied to linear systems and nonlinear systems of Lotka-Volterra type. This paper extends the domain of application to include the class of multivariable Lur'e-Postnikov systems, which plays a prominent role in transient stability studies of multimachine power systems.

2. THE INCLUSION PRINCIPLE AND STABILITY

Let us consider two dynamic systems

$$S\colon\ \dot{x} = f(t, x) \tag{2.1}$$

and

$$\tilde{S}\colon\ \dot{\tilde{x}} = \tilde{f}(t, \tilde{x})\ , \tag{2.2}$$

where $x(t) \in R^n$ and $\tilde{x}(t) \in R^{\tilde{n}}$ are the states of S and $\tilde{S}$ such that $n \leq \tilde{n}$. In (2.1) and (2.2), the functions $f\colon R \times R^n \to R^n$ and $\tilde{f}\colon R \times R^{\tilde{n}} \to R^{\tilde{n}}$ are assumed to be sufficiently smooth so that the solutions $x(t;\, t_0, x_0)$ and $\tilde{x}(t;\, t_0, \tilde{x}_0)$ of (2.1) and (2.2) exist and are unique for all initial conditions $(t_0, x_0) \in R \times R^n$ and $(t_0, \tilde{x}_0) \in R \times R^{\tilde{n}}$.

We introduce a linear transformation

$$\tilde{x} = Tx\ , \tag{2.3}$$

---

*The research reported herein has been supported by the U.S. Department of Energy, Electric Energy Division, under Contract EC-77-S-03-1493.

where T is an $\tilde{n} \times n$ constant matrix with full column rank. The inclusion principle [1] is stated as follows:

(2.4) DEFINITION. *We say that the system* $\tilde{S}$ *includes the system* S *if there exists a matrix* T *such that for any initial condition* (t_0, x_0) *of* S, $\tilde{x}_0 = Tx_0$ *implies*

$$x(t; t_0, x_0) = T^I\tilde{x}(t; t_0, \tilde{x}_0) \text{ for all } t \geq t_0 \tag{2.5}$$

where T^I *is a generalized inverse of* T .

If $\tilde{S}$ includes S it is called an *expansion* of S, and S is referred to as a *contraction* of $\tilde{S}$. These terms are justified by our assumption that the dimension of $\tilde{S}$ is larger than (or at least equal to) the dimension of S. Useful aspects of the inclusion, expansion, and contraction will be demonstrated in Section 4 when we deal with generalized decompositions of multimachine systems. At present, we note that if $\tilde{S}$ includes S, then any information about the motions of S can be extracted from the motions of $\tilde{S}$.

It is now of interest to derive conditions under which $\tilde{S}$ includes S. Using the matrix T of the linear transformation (2.3), we represent the function $\tilde{f}$ as

$$\tilde{f}(t, \tilde{x}) = T\, f(t, T^I\tilde{x}) + \tilde{m}(t, \tilde{x}) \tag{2.6}$$

where $\tilde{m}\colon R \times R^{\tilde{n}} \to R^{\tilde{n}}$ is called a *complimentary function.* For $\tilde{S}$ to include S, $\tilde{m}$ is required to satisfy restrictions, which are provided by the following [2]:

(2.7) THEOREM. $\tilde{S}$ *includes* S *if either*

$$\text{(i)} \quad \tilde{m}(t, Tx) = 0 \quad \text{for all} \quad (t, x) \in R \times R^n \tag{2.8}$$

or

$$\text{(ii)} \quad T^I\tilde{m}(t, \tilde{x}) = 0 \quad \text{for all} \quad (t, \tilde{x}) \in R \times R^{\tilde{n}} \quad . \tag{2.9}$$

We note that the two conditions in Theorem (2.7) do not imply each other. They are two independent conditions for $\tilde{S}$ to be an expansion of S, that is, for S to be a contraction of $\tilde{S}$.

We now turn our attention to stability properties of the two systems S and $\tilde{S}$. Because $\tilde{S}$ contains all the information about the motions of S, we expect to be able to extract stability properties of S from those of $\tilde{S}$. This we can do only after additional considerations are given to the equilibria of S and $\tilde{S}$. In fact, we require that the equilibrium of S is preserved in $\tilde{S}$ under the linear transformation (2.3). More precisely, if $f(t, x^*) = 0$ for all t and x^* is an equilibrium of S, then we demand that $\tilde{f}(t, Tx^*) = 0$ for all t so that $\tilde{x}^* = Tx^*$ is the corresponding equilibrium of $\tilde{S}$. It is easy to see that this takes place if and only if

$$\tilde{m}(t, Tx^*) = 0 \quad \text{for all} \quad t \in R \, . \tag{2.10}$$

The condition (2.10) is implied by the inclusion condition (i) of Theorem (2.7), but is not implied by the condition (ii).

The stability problem is resolved by the following:

(2.11) THEOREM. *Suppose that* $\tilde{S}$ *is an expansion of* S, *and* $\tilde{x}^* = Tx^*$. *Then stability of the equilibrium* $\tilde{x}^*$ *of* $\tilde{S}$ *implies stability of the equilibrium* x^* *of* S.

It is important to note that the converse of Theorem (2.11) is not true in general. It has been shown [1] that the converse can be established for linear systems provided certain additional conditions are satisfied.

In Section 4, when we consider multimachine power systems, stability is not a global property of the system. For this reason, it is of interest to establish the relationship between the regions of attraction $\Omega = \{x_0 \in R^n: x(t;\, t_0,\, x_0) \to x^*,\ t \to +\infty\}$ and $\tilde{\Omega} = \{\tilde{x}_0 \in R^{\tilde{n}}: \tilde{x}(t;\, t_0,\, \tilde{x}_0) \to \tilde{x}^*,\ t \to +\infty\}$ of the systems S and $\tilde{S}$. Definition (2.4) implies the following:

(2.12) THEOREM. *If* $\tilde{S}$ *includes* S *and* $\tilde{x}^* = Tx^*$, *then*

$$\Omega \supseteq \{x \in R^n: x = T^I\tilde{x},\ \tilde{x} \in \tilde{\Omega} \cap \text{Range } T\}\ . \tag{2.13}$$

Relation (2.13) can be used to compute estimates of the region Ω from the region $\tilde{\Omega}$. This indirect procedure is necessary when the systems are composed of overlapping interconnected subsystems and expansions are used to obtain an equivalent model in which the subsystems appear as disjoint. In this way, standard schemes of vector Liapunov functions [3] can be used to compute the region of attraction in the expanded state space and obtain the regions of attraction for the original system using Theorem (2.12). The procedure has been used implicitly in the studies of transient stability of multimachine power systems [4-8]. In Section 4, an explicit justification is provided.

3. LUR'E-POSTNIKOV SYSTEMS

In this section, we apply the inclusion principle to the multivariable version of Lur'e-Postnikov systems [9]. We consider a pair of systems $(S, \tilde{S})$ described by the equations

$$S:\ \begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \\ u &= \phi(y) \end{aligned} \tag{3.1}$$

$$\tilde{S}:\ \begin{aligned} \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}\tilde{u} \\ \tilde{y} &= \tilde{C}\tilde{x} \\ \tilde{u} &= \tilde{\phi}(\tilde{y}) \end{aligned} \tag{3.2}$$

where $x(t) \in R^n$ and $\tilde{x}(t) \in R^{\tilde{n}}$ $(n \leq \tilde{n})$ are the states of S and $\tilde{S}$; $u(t)$, $y(t) \in R^m$ and $\tilde{u}(t)$, $\tilde{y}(t) \in R^{\tilde{m}}$ $(m \leq \tilde{m})$. The matrices A, B, C and $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ are constant and of appropriate dimensions. The functions $\phi: R^m \to R^m$, $\tilde{\phi}: R^{\tilde{m}} \to R^{\tilde{m}}$ are given as

$$\begin{aligned} \phi(y) &= [\phi_1(y_1),\ \phi_2(y_2),\ \ldots\ ,\ \phi_m(y_m)]^T \\ \tilde{\phi}(\tilde{y}) &= [\tilde{\phi}_1(\tilde{y}_1),\ \tilde{\phi}_2(\tilde{y}_2),\ \ldots\ ,\ \tilde{\phi}_{\tilde{m}}(\tilde{y}_{\tilde{m}})]^T \end{aligned} \tag{3.3}$$

with sufficiently smooth components to ensure the existence and uniqueness of solutions of S and $\tilde{S}$. When we consider stability of the two systems S and $\tilde{S}$, we also assume that $x = 0$ and $\tilde{x} = 0$ are equilibrium states.

To provide conditions under which $\tilde{S}$ includes S, we introduce the linear transformations

$$\tilde{u} = Uu\ ,\quad \tilde{y} = Vy \tag{3.4}$$

where U and V are $\tilde{m} \times m$ matrices with full column rank. Applying the transformations (2.3) and (3.4) to S of (3.1) we get $\tilde{S}$ in (3.2) with

$$\tilde{A} = TAT^I + M , \quad \tilde{B} = TBU^I + N$$
$$\tilde{C} = VCT^I + L , \quad \tilde{\phi}(\tilde{y}) = U\phi(V^I\tilde{y}) + \tilde{\psi}(\tilde{y}) \tag{3.5}$$

where M, N, and L are *complementary matrices*, which have appropriate dimensions, and $\tilde{\psi}: R^{\tilde{m}} \to R^{\tilde{m}}$ is a complementary function. In (3.5), superscript I denotes the generalized inverse of the corresponding matrix.

We establish the following:

(3.6) THEOREM. *$\tilde{S}$ includes S if at least one of the following conditions holds:*

(i)
$$MT = 0 , \quad LT = 0 , \quad NU = 0$$
$$\tilde{\psi}(Vy) = 0 \quad \text{for all} \quad y \in R^m \tag{3.7}$$

(ii)
$$T^I M = 0 , \quad T^I N = 0 , \quad V^I L = 0$$
$$U^I\tilde{\psi}(\tilde{y}) = 0 \quad \text{for all} \quad \tilde{y} \in R^{\tilde{m}} \tag{3.8}$$

(iii)
$$T^I M^i T = 0 , \quad T^I M^{i-1} N = 0$$
$$LM^{i-1} T = 0 , \quad LM^{i-1} N = 0 , \quad i = 1, 2, \dots , \tilde{n}$$
$$U^I\tilde{\psi}(Vy) = 0 \quad \text{for all} \quad y \in R^m . \tag{3.9}$$

The conditions (i) and (ii) are consequences of Theorem (2.7). The condition (iii) is due to the additional information about the system structure provided in the class of Lur'e-Postnikov systems. All conditions in Theorem (3.6) are independent of each other.

The Inclusion Principle has been useful in the analysis of overlapping decompositions and stability in large-scale systems [1, 2]. When a system is composed of overlapping subsystems, an expansion can be formed in which the subsystems appear as disjoint. Then, standard techniques of vector Liapunov functions can be used to establish stability of the original system from stability of the expansion. Concrete examples of this methodology as applied to nonlinear systems has been provided in the context of multimachine power systems [8] and Lotka-Volterra models of multi-species communities [10]. In Section 4, we reconsider the power system example in the framework of the Inclusion Principle.

4. MULTIMACHINE SYSTEM

We start with the standard description of an N-machine power system in which the absolute motion of the i-th machine is described by the equation

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = P_{mi} - P_{ei} , \quad i = 1, 2, \dots , N \tag{4.1}$$

where

$$P_{ei} = E_i^2 G_i + \sum_{j \neq i}^{N} E_i E_j Y_{ij} \sin (\delta_i - \delta_j), \quad i = 1, 2, \dots , N \tag{4.2}$$

and

δ_i - the absolute rotor angle

M_i - the inertia coefficient

D_i - the damping coefficient

P_{mi} - the mechanical power delivered to the i-th machine

P_{ei} - the electrical power delivered by the i-th machine

E_i - the internal voltage

Y_{ij} - the transfer susceptance between the i-th and j-th machines

G_i - short-circuit conductance of the i-th machine.

In (4.1), it is assumed that M_i, P_{mi} and E_i are constant for all machines.

Let δ_i^o's be components of the post-fault locally stable equilibrium, which are determined by the equations

$$E_i^2 G_i + \sum_{j \neq i}^{N} E_i E_j Y_{ij} \sin(\delta_i^o - \delta_j^o) = P_{mi} \ , \quad i = 1, 2, \ldots , N \qquad (4.3)$$

where one of δ_i^o's is arbitrary. Substituting (4.2) and (4.3) into (4.1), we obtain

$$M_i \ddot{\delta}_i + D_i \dot{\delta}_i = -E_i \sum_{j \neq i}^{N} E_j Y_{ij} \left[\sin(\delta_i - \delta_j) - \sin(\delta_i^o - \delta_j^o)\right]$$

$$i = 1, 2, \ldots , N \ . \qquad (4.4)$$

Choosing the first machine as reference, we obtain the following state space description

$$\begin{aligned} S\colon \dot{x} &= Ax + Bu \\ y &= Cx \\ u &= \phi(y) \end{aligned} \qquad (4.5)$$

where $x(t) \in \mathcal{R}^{2N-1}$ is the state vector defined as

$$x_i = \omega_i \ , \quad i = 1, 2, \ldots , N$$

$$x_{N+j-1} = (\delta_1 - \delta_j) - (\delta_1^o - \delta_j^o) \ , \quad j = 2, 3, \ldots , N \qquad (4.6)$$

where $\omega_i = \dot{\delta}_i$ and $(\delta_1 - \delta_j) - (\delta_1^o - \delta_j^o)$ are velocity and angle differences with respect to the equilibrium. In (4.5), $u(t) \in \mathcal{R}^m$, $y(t) \in \mathcal{R}^m$ where $m = N(N-1)/2$, and the matrices A, B, and C are defined as

$$A = \left[\begin{array}{cc:c} \multicolumn{2}{c:}{-\Delta_N \Lambda_N^{-1}} & 0_{N,N-1} \\ \hdashline e_{N-1} & \multicolumn{1}{:c:}{-I_{N-1}} & 0_{N-1} \end{array}\right], \quad B = \left[\begin{array}{c} -\Lambda_N^{-1} J_{N,m} \\ \hdashline 0_{N-1,m} \end{array}\right], \quad C = \left[\begin{array}{c:c} 0_{m,N} & K_{m,N-1} \end{array}\right] \qquad (4.7)$$

with

$$J_{N,m} = \left[\begin{array}{c:c} e^T_{N-1} & o^T_{m-N+1} \\ \hdashline \multicolumn{2}{c}{-K^T_{m,N-1}} \end{array}\right], \quad K_{m,N-1} = \left[\begin{array}{c:c} \multicolumn{2}{c}{I_{N-1}} \\ \hdashline -e_{N-2} & I_{N-2} \\ \hdashline o_{N-3} & I_{N-3} \\ \hdashline \multicolumn{2}{c}{\cdots\cdots\cdots} \\ \hdashline 0 \;\; 0 \; \ldots \; 0 \; -1 & 1 \end{array}\right] \tag{4.8}$$

In (4.7) and (4.8), $e_{N-1} = (1, 1, \ldots, 1)^T \in R^{N-1}$, $o_{m-N+1} = (0, 0, \ldots, 0)^T \in R^{m-N+1}$, $\Lambda_N = \text{diag}\,\{M_1, M_2, \ldots, M_N\}$, $\Delta_N = \text{diag}\,\{D_1, D_2, \ldots, D_N\}$, I_{N-1} is the $(N-1) \times (N-1)$ identity matrix, and 0_{N-1}, $0_{N,N-1}$, $0_{N-1,N}$ zero matrices of indicated dimensions. The nonlinear functions $\phi(y)$ in (4.5) has the components

$$\phi_\ell(y_\ell) = \kappa_\ell[\sin(y_\ell + \delta_i^o - \delta_j^o) - \sin(\delta_i^o - \delta_j^o)]$$
$$\ell = 1, 2, \ldots, m \tag{4.9}$$

where $\kappa_\ell = E_i E_j Y_{ij}$, $\ell = \frac{1}{2}(2N-i)(i-1) + j - 1$, and $i = 1, 2, \ldots, N-1$; $j = i+1, i+2, \ldots, N$.

In order to simplify the stability analysis of the overall system, it is efficient to decompose the system S of (4.5) into several groups of machines so that each group forms an overlapping subsystem having in common the reference machine with the rest of the subsystems. The resulting subsystems are of the same Lur'e-Postnikov type but, of course, with a lower dimension. The resulting system, which is obtained by the decomposition of reference [8], can be reformulated to get the following compact form:

$$\begin{aligned} \tilde{S}\colon \dot{\tilde{x}} &= \tilde{A}\tilde{x} + \tilde{B}\tilde{u} \\ \tilde{y} &= \tilde{C}\tilde{x} \\ \tilde{u} &= \tilde{\phi}(\tilde{y}) \end{aligned} \tag{4.10}$$

where $\tilde{x}(t) \in R^{2N+s-2}$, $u(t)$, $y(t) \in R^m$, and s is the number of subsystems. In (4.10), the matrices are

$$\tilde{A} = \text{diag}\,\{\tilde{A}_1, \tilde{A}_2, \ldots, \tilde{A}_s\}$$
$$\tilde{B} = [\text{diag}\,\{\tilde{B}_1, \tilde{B}_2, \ldots, \tilde{B}_s\} \mid \tilde{B}_H]$$
$$\tilde{C} = \left[\begin{array}{c} \text{diag}\,\{\tilde{C}_1, \tilde{C}_2, \ldots, \tilde{C}_s\} \\ \hdashline \tilde{C}_H \end{array}\right]. \tag{4.11}$$

The matrices $\tilde{A}_k$, $\tilde{B}_k$, and $\tilde{C}_k$ have the same form as those of (4.7) with N and m replaced by N_k (the number of machines in the k-th subsystem) and $m_k = N_k(N_k-1)/2$. The specific forms of the matrices $\tilde{B}_H$ and $\tilde{C}_H$ can readily be obtained after a decomposition is chosen. Finally, we note that the nonlinear function ϕ can be obtained by reordering of the components of ϕ in (4.9).

Let us define the complementary function $\tilde{\Psi}$ as $\tilde{\Psi}(\tilde{y}) = 0$, and the following matrices:

$$T = \left[\begin{array}{c:c:c}
1 & o_{N-1}^T & o_{N-1}^T \\ \hdashline
o_{2(N_1-1)} & F_{11} & F_{12} \\ \hdashline
1 & o_{N-1}^T & o_{N-1}^T \\ \hdashline
o_{2(N_2-1)} & F_{21} & F_{22} \\ \hdashline
\vdots & \vdots & \vdots \\ \hdashline
1 & o_{N-1}^T & o_{N-1}^T \\ \hdashline
o_{2(N_s-1)} & F_{s1} & F_{s2}
\end{array}\right] , \quad T^I = (T^T T)^{-1} T^T$$

$$F_{i1} = \left[\begin{array}{c:c:c} 0_{2(N_i-1),N_1+\ldots+N_{i-1}-i+1} & \begin{array}{c} I_{N_i-1} \\ \hdashline 0_{N_i-1} \end{array} & 0_{2(N_i-1),N-N_1-\ldots-N_i+i-1} \end{array}\right]$$

$$F_{i2} = \left[\begin{array}{c:c:c} 0_{2(N_i-1),N_1+\ldots+N_{i-1}-i+1} & \begin{array}{c} 0_{N_i-1} \\ \hdashline I_{N_i-1} \end{array} & 0_{2(N_i-1),N-N_1-\ldots-N_i+i-1} \end{array}\right]$$

$$U = V^T , \; U^I = V ,$$

$$V = \left[\begin{array}{c}
\begin{array}{c:c} I_{N_1-1} & 0_{N_1-1,[N(N-1)-2N_1+2]/2} \end{array} \\ \hdashline
\begin{array}{c:c:c} 0_{N_1-2,N-1} & I_{N_1-2} & 0_{N_1-2,[(N-1)(N-2)-2N_1+4]/2} \end{array} \\ \hdashline
\begin{array}{c:c:c} 0_{N_1-3,2N-3} & I_{N_1-3} & 0_{N_1-3,[(N-1)(N-4)-2N_1+8]/2} \end{array} \\ \hdashline
\vdots \\ \hdashline
\begin{array}{c:c:c} o^T_{(N_1-2)(2N-N_1+1)/2} & 1 & o^T_{[(N-1)(N-2N_1+4)+(N_1-2)(N_1-3)-2]/2} \end{array} \\ \hdashline
\begin{array}{c:c:c} 0_{N_2-1,N_1-1} & I_{N_2-1} & 0_{N_2-1,[N(N-1)-2N_1-2N_2+4]/2} \end{array} \\ \hdashline
\begin{array}{c:c:c} 0_{N_2-2,N_1(2N-N_1-1)/2} & I_{N_2-2} & 0_{N_2-2,[(N-1)(N-2N_1)+N_1(N_1-1)]/2} \end{array} \\ \hdashline
\vdots \\ \hdashline
\begin{array}{c:c} o^T_{[N(N-1)-2]/2} & 1 \end{array} \\ \hdashline
\\
V_H
\end{array}\right]$$

$$V^I = V^T$$

$$M = \left[\begin{array}{c:c:c:c:c:c:c}
-[(s-1)/s]D_1M_1^{-1} & & (1/s)D_1M_1^{-1} & & & (1/s)D_1M_1^{-1} & \\
o_{N_1-1} & & o_{N_1-1} & & & o_{N_1-1} & \\
[(s-1)/s]e_{N_1-1} & & -(1/s)e_{N_1-1} & & & -(1/s)e_{N_1-1} & \\
(1/s)D_1M_1^{-1} & & -[(s-1)/s]D_1M_1^{-1} & & & (1/s)D_1M_1^{-1} & \\
o_{N_2-1} & & o_{N_2-1} & & & o_{N_2-1} & \\
-(1/s)e_{N_2-1} & & [(s-1)/s]e_{N_2-1} & & & -(1/s)e_{N_2-1} & \\
\vdots & 0_{2N+s-2,2(N_1-1)} & \vdots & 0_{2N+s-2,2(N_2-1)} & \cdots & \vdots & 0_{2N+s-2,2(N_s-1)} \\
(1/s)D_1M_1^{-1} & & (1/s)D_1M_1^{-1} & & & -[(s-1)/s]D_1M_1^{-1} & \\
o_{N_s-1} & & o_{N_s-1} & & & o_{N_s-1} & \\
-(1/s)e_{N_s-1} & & -(1/s)e_{N_s-1} & & & [(s-1)/s]e_{N_s-1} &
\end{array}\right]$$

$$N = 0\ , \quad L = 0 \tag{4.12}$$

where the submatrix V_H of V is a binary matrix which is determined by decomposition and such that the matrix V has exactly one unit element per column. The matrix V is a permutation matrix having the size $m \times m$.

To establish $\tilde{S}$ as an expansion of S with the above choice of matrices (4.12), we use the conditions (i) of Theorem (3.6) and (3.5). In the expansion $\tilde{S}$ the chosen groups of machines appear as disjoint subsystems, and the standard techniques of vector Liapunov functions can be used to compute estimates of the region of attraction for $\tilde{S}$. The estimates of the region of attraction for the original system S can be obtained using Theorem (2.12).

5. CONCLUSION

The concept of expansion and contraction has been broadened to include the nonlinear systems of Lur'e-Postnikov class. It has been shown how a lower order nonlinear system can be used to represent a higher dimensional nonlinear system of the same type. Alternatively, the concept of expansion can be used to represent a system in a larger state space where overlapping subsystems of the original system appear as disjoint. In the expanded space, it is possible to use standard methods for stability analysis of interconnected nonoverlapping subsystems. By using the Inclusion Principle, the stability properties of the expansion carry over to the original system. Moreover, the extent of the stability region can be extracted from the analysis of the expanded system. This expansion-contraction framework has been used to provide a complete account of the generalized decomposition scheme for transient stability analysis of multimachine power systems.

REFERENCES

[1] M. Ikeda and D. D. Šiljak: Overlapping decompositions, expansions, and contractions of dynamic systems. Large Scale Systems, vol. 1, pp. 29-38, 1980.

[2] M. Ikeda and D. D. Šiljak: Generalized decompositions of dynamic systems and vector Liapunov functions. IEEE Transactions (to appear in the Bellman Special Issue, Automatic Control, 1981).

[3] D. D. Šiljak: Large-Scale Dynamic Systems: Stability and Structure, North-Holland, New York, 1978.

[4] M. A. Pai and C. L. Narayana: Stability of large scale power systems, Proceedings of the 6th IFAC Congress, Boston, Massachusetts, pp. 31.6: 1-10, 1975.

[5] Lj. B. Jocić, M. Ribbens-Pavella, and D. D. Šiljak: Multimachine power systems: Stability, decomposition, and aggregation, IEEE Transactions, vol. AC-23, pp. 325-332, 1978.

[6] Lj. T. Grujić, M. Darwish, and J. Fantin: Coherence, vector Liapunov functions, and large-scale power systems. International Journal of Control, vol. 10, pp. 351-362, 1979.

[7] M. Araki, M. Saeki, and B. Kondo: Applications of a new stability condition of composite systems to multimachine power systems. IEEE Transactions, vol. AC-25, pp. 480-483.

[8] M. Araki, M. M. Metwally, and D. D. Šiljak: Generalized decompositions for transient stability analysis of multimachine power systems. Proceedings of JACC, Paper No. TA3-B, San Francisco, California, 1980.

[9] V. M. Popov: Hyperstability of Control Systems, Springer, New York, 1973.

[10] M. Ikeda and D. D. Šiljak: Lotka-Volterra equations: Stability, decomposition, and aggregation. Journal of Mathematical Biology, vol. 9, pp. 65-83, 1980.

DIRECT METHODS FOR OFF-LINE AND ON-LINE TRANSIENT STABILITY ASSESSMENT OF POWER SYSTEMS

M. RIBBENS-PAVELLA, J. SABATEL, A. CALVAER AND J. GHEURY

Department of El. Eng., University of Liège
Sart-Tilman B-28, B-4000 LIEGE BELGIUM

ABSTRACT

We derive a Liapunov-like criterion suitable for transient stability analysis, and use it in two different ways. The first is the conventional off-line one, consisting in evaluating a system's behaviour in terms of critical clearing times corresponding to important disturbances; the second way concerns a new methodology which avoids disturbance simulations and provide an appropriate tool for on-line stability assessment.

1. INTRODUCTION

The last two decades have seen the outburst of a new tool for transient stability analysis of power systems : the Liapunov direct method. Giving a list of references on the subject is beyond the scope of this paper (such a list may be found in survey papers, [1],[2]). Let us only say here that till the mid seventies, the investigations have essentially been directed along two distinct, although related, lines : the construction of suitable Liapunov functions and the definition of appropriate stability domain estimates. However, despite all the advances in the subject, there were still by the mid seventies, some remaining difficulties : not only did the construction of Liapunov functions impose the necessity of accounting for quite simplified mathematical models, but also the theoretical stability domain estimation appeared to provide quite conservative practical results. This has motivated further investigations focused on the definition of more suitable "practical" stability domain estimates relative to already constructed Liapunov functions, whereas at the same time another direct method, the so-called "vector Liapunov approach" started being explored. We shall not discuss this latter, since it has not as yet reached maturity; let us only say that by now it appears to be complementary rather than competitive with the "scalar" approach.

On the way towards obtaining less conservative practical results by the Liapunov criterion, investigations have been carried out, leading to the construction of larger stability domain estimates (SDEs). References [3 to 5] are among the most interesting works within this context. Another approach is the one proposed by our research group : based on a previous work [6], it has at first been reported in ref. [7], then improved in [8]. The first goal of this paper is to present this "practical" SDE (PSDE) and the derived "Liapunov-like" criterion. The quality of the practical results obtained by this means is tested via numerous stability simulations performed on five different systems. The simulations concern, as usual, important disturbances. The stability analysis is a conventional one, consisting in evaluating the severity of a given disturbance in terms of its critical clearing time. The proposed Liapunov-like criterion is shown to provide results generally in a quite good agreement with those of the numerical integration method; moreover, from a computing viewpoint, it appears to be very fast.

Besides being suitable for off-line stability evaluation, the aforementioned Liapunov-like approach possesses an even more interesting feature : the possibility of deriving a new methodology appropriate for on-line transient stability assessment. It consists in defining an index whose value is a measure of the stability behaviour of a power system in a given configuration, and whose computation is very fast. The second goal of the present paper is precisely to propose such a stability index, to derive an adequate expression and to outline its potentialities.

2. THE MATHEMATICAL MODEL

In the standard simplified description of an n-machine system, the absolute motion of the i-th machine is expressed by :

$$M_i\ddot{\delta}_i + D_i\dot{\delta}_i = P_{mi} - P_{ei} \qquad (1)$$

where :

δ_i : rotor angle of ith generator relative to a rotating reference

M_i, D_i : generator inertia and damping constants

P_{mi} : mechanical imput to ith generator from its turbine

P_{ei} : electrical output to the network.

The simple form of P_{ei} for constant generator voltage behind transient reactance is given by :

$$P_{ei} = E_i^2 Y_{ii} \cos\theta_{ii} + \sum_{\substack{j=1\\ j\neq i}}^{n} E_i E_j Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}) \qquad (2)$$

where :

E_i : ith generator internal voltage

Y_{ij}, θ_{ij} : modulus and argument of Y_{ij}, the ijth element of the admittance matrix $\underline{Y}$ reduced to the generators' nodes.

3. THE LIAPUNOV-LIKE CRITERION

3.1. Liapunov method

The energy-type Liapunov function[1] which has almost always been used in power system applications, necessitates adopting the following simplifying assumptions for the system descritption :

* constant E_i, P_{mi}, M_i

* $\theta_{ij} = \pi/2$ ($i\neq j$) for the system in its final configuration, i.e. neglecting transfer conductances in this configuration.

Linked to these assumptions, a primary requirement is the use of angles relative to a reference generator and, whenever, "uniform" damping is used, relative rotor velocities as well [1]. In this case, and when the nth generator acts as the reference, the equations of the system motion derive from expressions (1) and (2) (in the latter expression $(\delta_{ij} - \theta_{ij})$ is replaced by $\sin\delta_{ij}$) and take the form :

$$\dot{\delta}_{in} = \omega_{in} \qquad (i=1,2,\dots,n-1)$$

$$\dot{\omega}_{in} = \frac{D_i}{M_i}\omega_{in} + (P_{mi} - P_{ei})M_i^{-1} - (P_{mn} - P_{en})M_n^{-1}$$

$$= C\omega_{in} + f_{in}(\delta_{in}, \delta_{jn}) \qquad (i=1,2,\dots,n-1) \qquad (3)$$

where $\delta_{in} = \delta_i - \delta_n$; $\omega_{in} = \omega_i - \omega_n$.

---

[1] The term "V-function" will be used interchangeably with "Liapunov function" in this paper.

The solutions (or equilibrium points) of (3) :

$$\begin{array}{lcr} \omega^{\circ}_{in} = 0 & & \text{(a)} \\ & (i=1,2,\ldots,n-1) & \qquad (4) \\ f(\delta^{\circ}_{in},\delta^{\circ}_{jn}) = 0 & & \text{(b)} \end{array}$$

are therefore obtained by solving the (n-1) non-linear algebraic equations (4b). In what follows usage will be made of solutions of the system in its final configuration, in the state space of the (n-1) relative angles $\underline{\delta}$ and (n-1) relative speeds $\underline{\omega}$. We shall denote by $(\underline{\delta}^{s},\underline{0})$ the stable equilibrium point (SEP) -if any- and by $(\underline{\delta}^{ui},\underline{0})$ the ith unstable equilibrium point (UEP) surronding $(\underline{\delta}^{s},\underline{0})$.

The energy-type V-function linked to state equations (3) has the form [1]

$$V(\underline{\delta},\underline{\omega}) = V_k(\underline{\omega}) + V_p(\underline{\delta}) \qquad (5)$$

where k and p state for kinetic and potential energy respectively.

A SDE of this V-function may be theoretically determined by computing all UEPs surrounding the SEP and defining the boundary of the (closed) hypersurface V_{Th} [1] :

$$V_{Th} \triangleq \min_{i} V(\underline{\delta}^{ui},\underline{0}) . \qquad (6)$$

Obviously, such a computation would be unacceptably long for practical purposes due to the very large number of UEPs (from physical considerations one can propose 2^{n-1} starting points (SPs) for solving equations 4b). Much attention has therefore been paid to reducing the number of the a priori interesting UEPs to be considered. In ref. [6] we have suggested the consideration of only the n "type 1"(*) SPs for initiating the Davidon-Fletcher-Powell minimization procedure to solve equ. (4b). Further, based on the fact that the variation of V is small in the neighbourhood of a type 1 UEP, usage of these SPs rather than of their corresponding actual UEPs was suggested; this allows avoiding the minimization procedure.

All these improvements have contributed to speed up significantly the computation of the practical limit value of V which by this means is expressed by :

$$V_{\ell} = \min_{i=1,2,\ldots,n} V(\underline{\delta}^{si},\underline{0}) \qquad (7)$$

where $\underline{\delta}^{si}$ represents the ith type 1 SP.

However, both the theoretical (V_{Th}) and practical (V_{ℓ}) determinations of the SDE have the same property, that of making it depend on the system's final configuration; in other words, for a given system, the SDE is independent of the fault location (and also of the choice of the reference machine). Obviously, this is not physically justifiable; moreover, from a practical point of view the boundary V_{ℓ} generally leads to conservative results, especially in the case of systems comprising a large number of machines (see the numerical examples of Table I). As a matter of fact, this conservativeness is not surprising : it is well-known that the method provides in practice sufficient but not necessary stability conditions (for systems comprising more than two machines). Indeed, when the system motion is represented in the state space by a point located inside the limit hypersurface, the system is certainly stable; but it may still remain stable even if this point lies outside the closed SDE, provided that it does not approach a UEP [7].

3.2. Defining a "practical" SDE (PSDE)

Attempts to define larger SDEs have led to interesting studies, [3 to 5]. Although using different methodologies, all these studies aim to take account of the fault location by examining the way in which the system would separate

---

(*) These are points corresponding to the physical situation where one machine goes out of step whereas the others continue swinging together.

for a particular fault, and of the transfer conductances -at least an approximation of their effect- by introducing an integral term and integrating it along a (assumed) linearized trajectory. All these methods concentrate on the evaluation of a practical SDE at a type 1 UEP corresponding to "first swing step-out mode", based for that on a mixture of approximate, intuitive and theoretical methods. The results thus obtained seem quite satisfactory [5], despite some observed discrepencies (see discussion on [4]).

The procedure proposed in the present study is as well related to the machine(s) which at first loses (lose) its (their) synchronism during the transients; it differs, however, from the previous ones in the way of determining this (these) machine(s), and also in the way of evaluating the limit value of the V-function at the corresponding interesting UEP. The reasoning which has led us to the proposed procedure is the following. We consider as interesting limit value of the V-function the one taken by that open trajectory of the state space which, when followed by the system, makes it go far away from the SEP. This trajectory may in turn be related to the UEP corresponding to the situation "one machine (or a group of machines) starts going out of step whereas the others remain stable". Now, from a physical viewpoint this former is most likely to be the one possessing the "largest final acceleration", i.e. the largest acceleration at a "clearing time close to the actual critical clearing time". We shall consider as such the $t_{c\ell}$ provided by V_ℓ; indeed this $t_{c\ell}$ is equal to or (generally) smaller than the actual t_c (see numerical examples of Table I); it therefore constitutes an appropriate -and computationaly convenient- "clearing time close to the actual t_c".

On the basis of the above reasoning, we propose the following procedure leading to the evaluation of the so-called PSDE :

(i) Compute $t_{c\ell}$ corresponding to V_ℓ

(ii) At this $t_{c\ell}$ determine the machine(s) which possesses (possess) the maximum acceleration; let k denote this machine (or group of machines)

(iii) Consider as interesting SP the one corresponding to the situation "group of machines k going out of step; let $\underline{\delta}^{appr.}$ be this SP

(iv) Compute $V_L \triangleq V(\underline{\delta}^{appr.},\underline{0})$; (8)

this will be taken as the limit value of the V-function defining the PSDE.

The computation of the critical clearing time (CCT) corresponding to the above PSDE will be performed according to the well-known two-step scheme :

(α) by means of a numerical integration solve the equations of motion (3) of the system in its faulted configuration; compute the components of the vectors $(\underline{\delta}_e,\underline{\omega}_e)$ for successive fault clearing times, t_es

(β) compute the values taken by the V-function (5) at these t_es and compare them to V_L expressed by (8) untill reaching $V(\underline{\delta}_{cL},\underline{\omega}_{cL}) = V_L$; the corresponding t_{cL} will be the CCT provided by the Liapunov-like criterion.

3.3. Comments

1.- We have called the above-derived SDE "practical" to emphasize the fact that it is an open stability domain, contrary to the theoretical one; similarly, we have called "Liapunov-like criterion" the one using this PSDE to differentiate it from the Liapunov criterion which uses a closed SDE.

2.- By its very nature, the Liapunov-like procedure is computationally very fast (almost as fast as the one leading to the computation of V_ℓ).

3.- By construction, the above procedure leads to values V_L equal to or larger than V_ℓ; hence, $t_{cL} \geq t_{c\ell}$. This may lead to optimistic rather than pessimistic values of t_{cL}, contrary to $t_{c\ell}$ for which $t_{c\ell} \leq t_c$ (here t_c denotes the "actual" CCT i.e. that provided by an equivalent numerical integration procedure); thus, we may obtain $t_{cL} \lesssim t_c$.

4.- Because of neglecting transfer conductances, an inconsistency is introduced in the determination of the SEP of the system in its post-fault configuration[1]. In fact, we are faced with the following alternative :

---

[1] whose evaluation governs that of V_ℓ and of V_L.

- either compute the (presumed) stable solution of the system by solving the algebraic equations (4) where transfer conductances are neglected and where the initial values, P_{mi}s, are kept : but the solution thus obtained differs from the one given through a standard load flow (where the transfer conductances are taken into account); this inconsistency becomes obvious when the power system under study has the same configuration before the occurence of the fault and after the fault clearing : the stable pre-fault and post-fault solutions differ then from each other instead of coinciding as they should do;
- or re-compute new values, P'_{mi}s, so as to make the post-fault solution coincide with the one obtained by a standard load-flow program for this post-fault configuration[1]; but then we introduce an inconsistency by using different values of P_{mi}s for the disturbed (post-fault) and the undisturbed (pre-fault) system's motion.

As the phenomena are essentially non-linear, there is no a priori theoretical guide allowing to choose between these two possibilities. From a physical viewpoint, however, we can say that it is preferable to adopt the second solution, i.e. to recalculate new P'_{mi}s; otherwise we get a post-fault configuration very different from the pre-fault one. This is especially important for large systems, such as the 40-machine system considered below, whereas for small systems the difference between the two configurations is generally negligible. The above reason has led us to always re-compute new P'_{mi}s.

3.4. Illustrative examples.

We have considered different power systems comprising 3,7,9,15 and 40 machines, and have performed many simulations on each of them (one at a time). Due to obvious space limitations, the data and configuration of the systems are not reported here; they may be available upon request. Let us only note that the so-called 40-machine system is the 380-220-150 kV equivalent belgian system; its interconnections with the remaining european power system are replaced by external equivalents. Let us also observe that the 15-machine system has a symmetrical structure : the results concerning 7 of its machines are identical to those obtained for their corresponding symmetrical machines.

The considered disturbances are 3-ϕ short-circuits applied either at generators' or at loads' buses. The stability simulations are carried out in the standard way consisting in searching the CCT corresponding to each disturbance. The computation of the t_cs by the numerical integration procedure has been performed on the basis of the machines' swing curves explored during 1.5 second.

The results of all simulations are listed in Table I; its columns provide the following information :

- column 2 identifies the bus-bar at which the disturbance occurs and specifies whether the final configuration is identical to (I) or different from (D) the prefaulted one. "D : x,y" means : "different; tripped line : x,y";
- column 3 indicates the "actual" CCT provided by a step-by-step numerical integration (t_c);
- column 4 indicates the corresponding $t_{c\ell}$ provided by the limit value V_ℓ (Liapunov criterion) and also the relative difference between t_c and $t_{c\ell}$:

$$r_\ell = \frac{t_c - t_{c\ell}}{t_c} \%$$

- column 5 lists the same as above information provided by the limit value V_L (Liapunov-like criterion).

---

[1] in other words :
(i) run a load flow program with given $P_{mi}=P^\circ_{mi}$ to find a stable solution, $\underline{\delta}^s$
(ii) with this $\underline{\delta}^s$, solve equations (4) to determine new P'_{mi}s

TABLE I : COMPARISON OF CRITICAL CLEARING TIMES

1	2		3	4		5	
System	Fault Location		Step-by-Step	Liapunov		Final acceleration	
	BUS	Final config.	t_{cr} (10^{-2}s)	$t_{c\ell}$(10^{-2}s)	r_{ℓ}(%)	t_{cL}(10^{-2}s)	r_L(%)
3 - Mach.	1	I	40-41	36-37	10	36-37	10
	2	I	33-34	30-31	9	33-34	0
	3	I	19-29	17-18	10	19-20	0
	3	D : 3-4	16-17	17-18	- 6	19-20	-18
	6	D : 5-6	27-28	26-27	4	30-31	-11
	6	I	27-28	26-27	4	30-31	-11
7 - Mach.	1	I	35-36	26-27	26	35-36	0
	2	I	41-42	35-36	15	36-37	12
	3	I	39-40	27-28	31	39-40	0
	4	I	50-51	30-31	40	47-48	6
	5	I	35-36	26-27	26	35-36	0
	6	I	52-53	39-40	25	50-51	4
	6	D : 6-8	51-52	39-40	23	50-51	2
	8	D : 6-8	44-45	49-50	-11	49-50	-11
9 - Mach.	11	D : 11-13	53-54	39-40	26	58-59	- 9
	15	I	47-48	32-33	32	40-41	15
	3	I	26-27	21-22	19	27-28	- 4
	22	D : 22-20	92-93	74-75	20	81-82	12
	24	I	73-74	53-54	27	67-68	8
	26	I	36-37	27-28	25	35-36	3
15 -Mach.	Q1	I	41-42	28-29	32	38-39	7
	M1	I	36-37	33-37	8	38-39	- 6
	C1	I	41-42	38-39	7	38-39	7
	01	I	41-42	28-29	32	38-39	7
	X2	D : W2-X2	67-68	52-53	22	60-61	10
40 -Mach.	MERC 1	I	26-26	8-9	68	20-21	20
	AWIR 3	I	24-25	10-11	58	24-25	0
	COO 1	I	32-33	15-16	53	31-32	3
	GOUY 3	D : GOUY3-MONC3	31-32	10-11	68	23-24	26
	STAL 3	D : BERI3-STAL3	29-30	14-15	52	32-33	10
	AWIR 4	I	23-24	14-15	39	26-27	-13
	BRES 4	I	24-25	15-16	38	27-28	-12
	DOELIB	I	25-26	8-9	68	22-23	12
	VILG 4	I	25-26	17-18	32	28-29	-12
	FARC 4	I	27-28	22-23	18	28-29	- 4
	LANB 3	I	28-29	12-13	57	22-23	21
	SCHE 3A	I	30-31	12-13	60	22-23	27
	TRIV 3	I	32-33	13-14	59	22-23	31
	SERA 2	I	32-33	14-15	56	22-23	31
	GRAM 1	I	32-33	9-10	72	20-21	37

3.5. Discussion.

The results of Table I yield quite interesting comparisons :

* The Liapunov-like criterion is not anymore systematically conservative : there are also negative r_Ls (whereas all r_ℓs are positive).

* The absolute values of r_L are on the whole much smaller than the corresponding r_ℓs; this is even more noticeable for systems comprising a large number of machines; for instance, the r_ℓ values of the 40-machine system becomes so large that the Liapunov criterion becomes meaningless; on the contrary the Liapunov-like criterion is -apart from some few exceptions- in a good agreement with the step-by-step method.

* Compared with this latter method, the Liapunov-like criterion allows an appreciable gain of computing time, especially when the number of system generators increases : indeed, the calculation times necessitated for the above simulations range from 0.67/50 sec. for the 7-machine system, to 1.4/104, 4.8/170 and 36/525 for the 9,15 and 40-machine systems respectively, where the first value is by Liapunov and the second by the step-by-step procedure. Note that the gain in computing time could be much more important if particular efforts for computational efficiency were made. Note also that a further means to speed up computations consists in replacing the step-by-step integration over the fault trajectory by a limited Taylor expansion, provided the fault duration in less than about 0.5 sec. [6]. With all these improvements, the computing time required by the Liapunov-like criterion may be practically reduced to that imposed by the programs of the load-flow and of the reduction of the admittance matrix.

In addition to the results listed in Table I we have explored the following conjecture (which form a physical viewpoint sounds quite logical).

* *For a given power system and a given disturbance, the machine(s) possessing the maximum final acceleration is (are) also the one(s) which the first goes (go) out of step.* This conjecture has been verified in all cases except in the following two ones :

- in the 7-machine system, for mf=4 : mu=5
- in the 9-machine system, for fm=8 : mu=9

where mf and mu denotes respectively

mf : the machine which possesses the maximum final acceleration

mu : the machine which the first goes out of step (becomes unstable).

Another quite interesting property has also been verified in all cases :

* *For a given power system and a 3-ϕ short-circuit acting at a machine's bus-bar, the machine possessing the maximum final acceleration has without exception coincided with the one possessing the maximum initial acceleration* (by initial acceleration we obviously mean that appearing when the disturbance starts acting). Denoting by :

mi : the machine which possesses the maximum initial acceleration, the following identity has always been verified :

$$mf \equiv mi$$

However, when the disturbance is applied elsewhere than at a machine's bus-bar the above rule may fail During our simulations the following two exceptions have been found

- in the 9-machine system, for a 3-ϕ short-circuit at 24, mf=7, mi=3, mu=7
- in the 15-machine system, for a 3-ϕ short-circuit at X2, mf=M2, mi=N2, mu=K2.

It is to be noticed, however, that both exceptions correspond to disturbances with relatively large CCTs, i.e. to relatively small disturbances; physically, this is quite understable.

4. ON-LINE TRANSIENT STABILITY ASSESSMENT

In our opinion, the most important benefit that one may derive from the Liapunov-like criterion lies in the concept of new methodologies, different from the conventional transient stability analysis and appropriate for on-line stability evaluation. The concept proposed here constitues a "worst-case" assessment, based on the definition of a transient stability index (TSI). This is a scalar expression whose computation is very fast and whose value constitutes a measure of a system's robustness.

4.1. Concept and definition of TSI.

We relate TSI to the most dangerous disturbance, i.e. the one causing the most important instability in a given power system; in other words, we relate it to the disturbance leading to the smallest CCT. According to the Liapunov-like criterion developed in section 3, this disturbance is also the one providing the smallest V_L.

Based on the above considerations we now derive the following reasoning.

(a) The experience shows and we shall accept that *the most severe disturbance* that a power system may be subjected to is a 3-ϕ short-circuit at a machine's bus-bar.

(b) According to the Liapunov-like criterion, the value of the CCT of a given disturbance is proportional to V_L which in turn depends on the UEP relative to the machine possessing the *"largest final acceleration"* during the transients (as defined in section 3.2).

(c) It has been shown (section 3.5) that *whenever a 3-ϕ short-circuit acts at a machine's bus-bar,* the machine possessing the "largest final acceleration" coincides with the one possessing the *largest initial acceleration;* by "initial acceleration" we mean the acceleration taken by a machine at t_{o^+}, i.e. at the time at which the disturbance starts acting.

From considerations (a), (b) and (c) we conclude that the *most dangerous disturbance will be related to the machine possessing the largest initial acceleration when a 3-ϕ short-circuit acts at a (the same or another) machine's bus-bar.* A measure of the severity of a given disturbance may therefore be related to the difference between the maximum initial acceleration and a mean acceleration of the over-all system. Furthemore, a measure of the severity of the most dangerous disturbance may then be related to the maximum value among the aforementioned differences. The expression of TSI follows :

$$\eta \triangleq \max_{i=1,\dots,n} \Delta_k \tag{9}$$

where Δ_k denotes the above-mentioned difference :

$$\Delta_k = \gamma_{max} - \bar{\gamma} \tag{10}$$

where $\bar{\gamma}$ denotes the mean acceleration [9] and where the initial acceleration of the ith machine derives from equ. (1) and is expressed by :

$$\gamma_i = \left. \frac{P_{mi} - P_{ei}}{M_i} \right|_{t = t_{o^+}} \tag{11}$$

4.2. Practical implementation of TSI

According to its definition, calculating η implies simulating a 3-ϕ short-circuit at the bus-bar of successively each of the system machine. Note that this calculation does not necessitate the knwoledge of the system motion during a fault-on period. Hence, the only time consuming computation is imposed by equ.(11), i.e. by the computation of the admittance matrix reduced to the system's generators when one of them is subjected to a 3-ϕ short-circuit (see equ.(2)); consequently, there are in principle n such matrix reductions to be calculated. However, owning to the superposition technique used in ref. [9], these n matrix reductions may be replaced by only one and by some additional computations of simple analytic expressions; this allows a substantial saving of computing time.

An important feature of TSI is that although deriving from the Liapunov criterion, it does not depend on Liapunov functions and therefore does not imply accounting for simplifying assumptions currently imposed by them.

4.3. Practical use of TSI.

The suggested way of using the above TSI in practice is the following : assuming the value of TSI of a given system in its "safe" ("base case") configuration to be known, the "measure" of the stability of this system in its actual configuration (differing from the safe one because of any change of topology, generation power level, etc.) will be provided by :

$$\tau = \frac{\eta - \eta_s}{\eta_s} \tag{12}$$

where s states for "safe". This way of evaluating a system's reliability presumes a good knowledge of the system in order to determine η_s and also of a threshold of τ, above which the system should be considered in an emergency state.

4.4. Illustrative examples.

We have performed a large number of tests; some of them are reported in ref. [9], some others may be available upon request. For obvious reasons of space limitations we don't reproduce them here. Let us only say that several tests have been carried out with five power systems (the same as in section 3.4), with essentially the following two objectives :

* in order to check whether TSI constitutes a suitable measure of a system's robustness from its transient stability point of view;

* in order to evaluate the required time for its computation.

The results of these tests are very satisfactory : not only does the TSI behave properly, but also its calculation is quite fast : the required computing times have been found to range from 0.536 sec. for the 7-machine system, to 0.804, 1.865 and 14.305 for the 9, 15 and 40-machine systems respectively. Although quite short, these times are in fact much more important than those which would actually be necessitated in a current practical use of TSI; indeed, the main part of these times (more than 90%) have been required to run the $\underline{Y}$ matrix reduction algorithm; using an efficient programming for this latter would allow a substantial gain of computing times.

5. CONCLUSIONS

A practical stability domain estimate (PSDE) has been defined and used in two different approaches for transient stability evaluation.

The one concerns the standard stability analysis, quantified by means of fault critical clearing times. Based on the above PSDE, we have derived and tested a Liapunov-like criterion. The quality of the numerous practical results thus obtained shows that the proposed practical method is reliable. Moreover, its rapidity with respect to the numerical integration procedure should justify using it preferably to the latter, especially for a first evaluation; during short intervals, indeed, the simplified mathematical model may be considered sufficiently accurate.

The second way concerns the use of a new concept appropriate for on-line evaluation of a system's robustness. On the basis of the Liapunov-like criterion such a concept has been set forth and the analytic expression of the derived stability index has been constructed. If the ability of this index to detecting the most critical situations has by now been established, further research work is however needed in order to assess its efficiency in practical applications, especially its sensitivity to system parameters. This kind of investigations are at present pursued by our research group.

6. REFERENCES

[1] Ribbens-Pavella, M., "Critical survey of transient stability studies of multi-machine power systems by Liapunov's direct methods", Proc. of the 9th Annual Allerton Conference on Circuits and Systems Theory, pp. 751-767, October 1971. See also Ribbens-Pavella, M., "Transient stability of multimachine power systems by Liapunov's direct method", IEEE Winter Power Meeting, New York, Jan. 1971.

[2] Fouad, A.A., "Stability theory-criteria for transient stability", Engineering Foundation Conference Systems Engineering for Power, Henniker, New Hampshire, 1975, pp. 421-453.

[3] Saito, O., Koizumi, K., Volo, M., Mukae, H., Murotani, K., "Security monitoring systems including fast transient studies", IEEE Trans. on Power Apparatus and Systems, 1975, PAS-94, pp. 1789-1806.

[4] Athay, T., Podmore, R., Virmani, S., "A practical method for the direct analysis of transient stability", IEEE Trans. Power Apparatus and Systems, Vol. PAS-98, n°2, pp. 573-581, March/April 1979. See also Athay, T. Sherkat, V.R., Podmore, R., Virmani, S. and Puech, C., Final Report "Transient Energy Stability Analysis", presented at the Engineering Foundation Conference, U.S. Department of Energy, Sept. 30-Oct.5 1979, Davos, Switzerland.

[5] Kakimoto, N., Ohsawa, Y. and Hayashi, M., "Transient stability analysis of multimachine power systems with field flux decays via Liapunov's direct method", A 80 231-1, IEEE PES Winter Meeting, Feb. 3-8 1980.

[6] Ribbens-Pavella, M., Lemal, B., "Fast determination of stability regions for on-line transient power system studies", Proc. IEE, Vol. 123, n°7, pp. 689-696, July 1976. See also Ribbens-Pavella, M., Lemal, B. and Pirard, W., "On-line operation of Liapunov criterion for transient stability studies", IFAC Symposium 1977, Melbourne, 21-25 Feb., pp. 292-296.

[7] Ribbens-Pavella, M., Horward, J.L., Lemal, B., Nguyen, T.Ph., "Transient stabiltiy analysis by scalar Liapunov functions : recent improvement and practical results", Collection des Publications de la Faculté des Sciences Appliquées de l'Université de Liège, n°2, pp. 76-100, 1977.

[8] Gheury, J., "Critères de stabilité par méthodes directes", Travail de fin d'études, Université de Liège, May 1979.

[9] Ribbens-Pavella, M., Calvaer, A. and Gheury, J., "Transient stability index for on-line evaluation", IEEE PES Winter Meeting, New York, NY, Feb. 3-8, 1980, paper A 80 013-3.

SIGNAL DETECTION BASED ON DISCRETE DATA

C.C. LEE
Dept. of Elect. Eng. & Com. Sci.
Northwestern University
Evanston, Illinois 60201

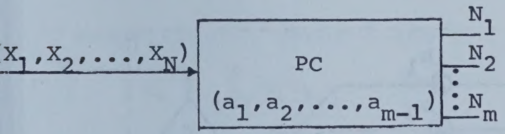
J.B. THOMAS
Dept. of Elect. Eng. &
Computer Science
Princeton University
Princeton, N.J. 08544

ABSTRACT

In this paper we transform the input data and their statistical characteristics into a multinomial vector and a set of multinomial parameters before designing an appropriate detection scheme. The Neyman-Pearson optimal detector and a likelihood ratio detector are derived and analyzed, for the case where the multinomial parameters are available under the alternative and for the case where they are not. These detectors are simple, flexible and their performance can be quite good.

I. INTRODUCTION

This paper considers detectors whose designs are based on observing a finite number of integer-valued random variables resulting from operating on the input data with a partitioner-counter type of device (denoted by "PC") shown in Fig. 1. Instead of characterizing statistically the input data itself, we consider the output from the PC. The major advantage of doing this is that the probability distribution of the PC's output is always multinomial whatever the actual distribution of the input data. Since a multinomial distribution can be completely indexed by a finite number of real parameters, a detector can be designed from an estimation of these parameters. The resulting detector is then nonparametric in the sense that a complete knowledge (i.e., the functional form of the noise density) is not necessary for the design of the detection scheme.



N_k = no. of X_i with $a_{k-1} < X_i < a_k$

$a_0 = -\infty, \qquad a_m = \infty$

Fig. 1. A Partitioner-Counter (PC) type of device of order m

A general analysis and discussion are given for a class of detectors whose test statistics are linear combinations of the PC's output. The weighting coefficients are assumed to be independent of the PC's output. For small sample size, a generating function is developed by which the frequency function of the test statistic can be obtained. For large sample size, the Gaussian approximation is employed to formulate the asymptotic distribution of the test statistic. Also, the efficacy of the test statistic is derived. A special subset of this class of detectors, called "nonparametric detectors based on m-interval partitioning", was proposed and investigated by Ching and Kurz (1). For these detectors, the PC device was so designed that all multinomial parameters under the hypothesis are equal. The sign detector and its generalizations [2,3] form another subset of this class of detectors.

Then, given an appropriately designed PC, the Neyman-Pearson optimal detection scheme based on observing the multinomial vector is derived for the special case where the multinomial parameters are available under both the hypothesis and the alternative when detection starts.

In general, the multinomial parameters under the alternative are unknown since we usually cannot predict accurately the signal strength. In this case, the Neyman-Pearson optimal detector does not exist and a level -α likelihood ratio detector is derived and investigated instead. The test statistic of this likelihood ratio detector under the hypothesis is shown to be approximately equal to the Pearson's chi-square test statistic and has asymptotically (for a large number or samples) a chi-square distribution under the hypothesis and a noncentral chi-square distribution under the alternative. The test threshold and the power function are obtained from these approximation techniques and asymptotic properties.

The PC device can be characterized by a finite number of parameters. Unfortunately, there does not exist a best choice of these parameters. Some reasonable and convenient structures including the "m-interval partitioning" [1] are suggested for the design of the PC. Based on these structures, we investigate, by simulations, the performances of the Neyman-Pearson optimal detector and the likelihood ratio detector which we have derived. If we consider these detectors and the PC device as a whole, these detectors are simple in structure, flexible in noise environment, and moderately good in performance.

II. A CLASS OF NONPARAMETRIC DETECTORS BASED ON DISCRETE DATA

Consider the class of detectors defined in Fig. 2, in which the function of the PC is described in Fig. 1. Each choice of the parameters $\{a_i\}$ and $\{b_i\}$ define an individual detector in the class. For simplicity, we assume that the parameters $\{b_i\}$ are independent of the PC's output.

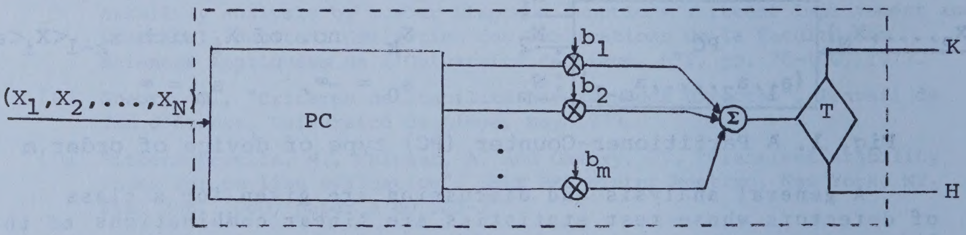


Fig. 2. Block diagram of a class of nonparametric detectors

Now, consider the test statistic used,

$$T_N = \sum_{k=1}^{m} b_k N_k \qquad (1)$$

It is apparent that $(N_1, N_2, \ldots N_m)$ has a multinomial distribution with parameters $(N; \theta_1, \ldots . \theta_m)$ under H and $(N; \eta_1, \ldots \eta_m)$ under K where

$$\theta_k \triangleq \Pr\left\{a_{k-1}<X_i<a_k \middle| H\right\} \tag{2}$$

and $$k=1,2,\ldots m$$

$$\eta_k \triangleq \Pr\left\{a_{k-1}<X_i<a_k \middle| K\right\} \tag{3}$$

Therefore, we have that

$$\sum_{k=1}^{m} N_k = N, \ \sum_{k=1}^{m} \theta_k = 1, \text{ and } \sum_{k=1}^{m} \eta_k = 1 \tag{4}$$

The exact distribution of T_N depends on the weighting coefficients $\{b_k\}$. In principle, it can be obtained with the aid of the following generating function:

$$g(x) \triangleq \sum_{r \in R} \Pr\{T_N = r\} \cdot x^r = \left[\sum_{k=1}^{m} p_k x^{b_k}\right]^N \tag{5}$$

where $p_k = \theta_k$ under H and $P_k = \eta_k$ under K and R extends over all possible values of T_N and is a finite set containing no more that m^N elements. When N is small, this can be done without too much difficulty. However, such a discrete frequency function becomes too complicated to obtain when N gets large. Fortunately, this difficulty can be circumvented by an alternative approach discussed below. Since $\sum_{k=1}^{m} N_k = N$, T_N can be written as

$$T_N = \sum_{i=1}^{N} c_i \tag{6}$$

where $(c_1, c_2, \ldots c_N)$ are i.i.d. random variables with

$$\Pr\left\{c_i = b_k \middle| H\right\} = \theta_k, \quad \Pr\left\{c_i = b_k \middle| K\right\} = \eta_k, \qquad k=1,2,\ldots m \tag{7}$$

The mean and the variance of each c_i are thus given by

$$\mu_c = \sum_{k=1}^{m} b_k \theta_k \qquad \text{and} \qquad \sigma_c^2 = \left(\sum_{k=1}^{m} b_k^2 \theta_k\right) - \mu_c^2 \tag{8}$$

respectively, under H and by

$$\mu'_c = \sum_{k=1}^{m} b_k \eta_k \qquad \text{and} \qquad \sigma'^2_c = \left(\sum_{k=1}^{m} b_k^2 \eta_k\right) - \mu'^2_c \tag{9}$$

respectively, under K. Therefore, the mean and the variance of T_N are given by

$$\mu_{T_N} = N\mu_c \qquad \text{and} \qquad \sigma^2_{T_N} = N\sigma_c^2 \tag{10}$$

respectively, under H and by

$$\mu'^{2}_{T_N} = N\mu'_c \qquad \text{and} \qquad \sigma'^2_{T_N} = N\sigma'^2_c \tag{11}$$

respectively, under K. When N is sufficiently large, T_N is asymptotically Gaussian distributed: that is,

$$\Pr\left\{(T_N - \mu_{T_N})/\sigma_{T_N} \le z\right\} \rightarrow \Phi(z) \tag{12}$$

under H and

$$\Pr\left\{(T_N-\mu'_{T_N})/\sigma'_{T_N}\leq z\right\}\to\Phi(z) \tag{13}$$

under K where $\Phi(\cdot)$ is the standardized Gaussian distribution function.

Upon obtaining the distribution of T_N (exactly or approximately), the test threshold required by a level-α test can be determined, and the power function can be calculated.

In the problem of detecting a constant signal, we have that

$$\theta_k = F(a_k) - F(a_{k-1}) \text{ and } \eta_k = F(a_k-s) - F(a_{k-1}-s) \tag{14}$$

where $F(\cdot)$ represents the noise distribution function and s denotes the signal strength. Thus, the efficacy [4] of T_N can be obtained [9]:

$$\epsilon(T_N) = \left\{\sum_{k=1}^{m} b_k\left[f(a_{k-1})-f(a_k)\right]\right\}^2 \Big/ \left\{\sum_{k=1}^{m} b_k^2\theta_k-\left[\sum_{k=1}^{m} b_k\theta_k\right]^2\right\} \tag{15}$$

where θ_k is given by Eq. (14) and $f(\cdot)$ represents the noise density function. This efficacy enables us to compare the class of detectors under investigation to other detectors in terms of the Pitman's asymptotic relative efficiency [4,5].

Although specific criteria can result in specific selection of the parameters m, $\{a_k\}$, and $\{b_k\}$, it should be noted that all the discussion and analysis given in this section are generally applicable to the whole class of detectors defined in Fig. 2 no matter how these parameters are chosen.

III. THE NEYMAN-PEARSON OPTIMAL DETECTOR BASED ON THE MULTINOMIAL VECTOR $(N_1,N_2,\ldots,N_m)$ - $\{\eta_k\}$ AVAILABLE

In this section, we will consider the detection problem of a constant (not necessarily known) signal in additive white noise. We assume that

(i) the actual noise distribution is arbitrary.
(ii) there are in total n input samples available,
(iii) the PC has been appropriately designed, so that the parameters $\{a_k\}$ are given,
(iv) the parameters $\{\theta_k\}$ have been estimated and are thus available when the detection starts, and
(v) the parameters $\{\eta_k\}$ are also available (This is true in some speical cases; for example, when both the functional form of the noise density and the signal strength are known before we start the detection)

Then, based on $(N_1,N_2,\ldots,N_m)$, the output of the PC, we can derive the Neyman-Pearson optimal detector. This problem is essentially that of searching for the uniformly most powerful (UMP) test in testing a simple hypothesis against a simple alternative:

$$H: \underline{p} = \underline{\theta} \qquad \text{vs.} \qquad K: \underline{p} = \underline{\eta}$$

upon observing a multinomial vector $(N_1,N_2,\ldots,N_m)$ with parameters n and $\underline{p}$ where $\underline{p} = (p_1,p_2,\ldots,p_m)$, $\underline{\theta} = (\theta_1,\theta_2,\ldots,\theta_m)$, and $\underline{\eta} = (\eta_1,\eta_2,\ldots\eta_m)$. Since the likelihood ratio is given by

$$L(N_1,\ldots N_m) = \frac{p(N_1,N_2,\ldots N_m;\ \underline{\eta})}{p(N_1,N_2,\ldots N_m;\ \underline{\theta})} = \frac{\frac{n!}{N_1!\ldots N_m!}\ \eta_1^{N_1}\eta_2^{N_2}\ldots\eta_m^{N_m}}{\frac{n!}{N_1!\ldots N_m!}\ \theta_1^{N_1}\theta_2^{N_2}\ldots\theta_m^{N_m}}$$

or

$$\log L(N_1,\ldots,N_m) = \sum_{k=1}^{m} N_k \log \frac{\eta_k}{\theta_k} \qquad (16)$$

the Neyman-Pearson optimal detection scheme is the following

$$T^*_n = \sum_{k=1}^{m} N_k \log \frac{\eta_k}{\theta_k} \begin{cases} > \\ < \end{cases} t(\alpha) \begin{matrix} \Rightarrow K \\ \Rightarrow H \end{matrix} \qquad (17)$$

where $t(\alpha)$ is the test threshold for a level-α test. We conclude that the optimal values of the detector parameters $\{b^*_k\}$ are given by

$$b^*_k = \log(\eta_k/\theta_k) \qquad (18)$$

To obtain $t(\alpha)$, we have to investigate the probability distribution of the test statistic T^*_n. If n is small, as indicated in Section II, the frequency function to T^*_n can be obtained from the generating function of Eq.(5). Denote the function thus obtained by $h(\cdot)$ under H. Then, $t(\alpha)$is given by

$$t(\alpha) = \min\left\{t \,\middle|\, \sum_{u \geq t} h(u) \leq \alpha\right\} \qquad (19)$$

If n is large, in parallel with (12) and (13), we have that

$$\Pr\left\{\frac{(T^*_n - \mu_{T^*_n})}{\sigma_{T^*_n}} \leq z \rightarrow\right\}\Phi(z) \quad \text{and} \quad \Pr\left\{\frac{(T^*_n - \mu'_{T^*_n})}{\sigma'_{T^*_n}} \leq z\right\}\rightarrow \Phi(z) \qquad (20)$$

under H and under K respectively, where

$$\mu_{T^*_n} = n \sum_{k=1}^{m} \theta_k b^*_k, \qquad \mu'_{T^*_n} = n \sum_{k=1}^{m} \eta_k b^*_k \qquad (21)$$

$$\sigma^2_{T^*_n} = n\left[\sum_{k=1}^{m} \theta_k b^{*2}_k - \left(\sum_{k=1}^{m} \theta_k b^*_k\right)^2\right] \qquad (22)$$

and

$$\sigma'^2_{T^*_n} = n\left[\sum_{k=1}^{m} \eta_k b^{*2}_k - \left(\sum_{k=1}^{m} \eta_k b^*_k\right)^2\right] \qquad (23)$$

Therefore, when n is large, the test threshold of the level-α Neyman-Pearson optimal detector is approximately given by

$$t(\alpha) = \mu_{T^*_n} + \sigma_{T^*_n}\ \Phi^{-1}(1-\alpha) \qquad (24)$$

where $\Phi^{-1}(u)$ dentoes the u-th quantile of the standardized Gaussian distribution. The power function can now be obtained:

$$\beta = 1 - \Phi\left\{\left[t(\alpha) - \mu'_{T^*_n}\right] / \sigma'_{T^*_n}\right\} \qquad (25)$$

The asymptotic efficiency of the resulting detector can be measured in terms of the efficacy of T_n^*. By definition [4], we have to investigate the limit case as the signal vanishes ($K \rightarrow H$) when we study the efficacy of a test statistic. Since those b_k^*'s obtained in Eq. (18) involve η_k's which depend on the signal strength (Eq. (14)), we have to use $\bar{b}_k \triangleq \lim_{s \to 0} b_k^*$ instead of b_k^* when we apply Eq. (15) to calculate the efficacy of T_n^*. Since

$$F(x-s) \rightarrow F(x) - sf(x) \quad \text{as } s \rightarrow 0$$

if $f(\cdot)$ is continuous, it follows from Eq. (14) that

$$(\eta_k/\theta_k) \rightarrow 1 + s\left[f(a_{k-1}) - f(a_k)\right]/\theta_k \qquad \text{as } s \rightarrow 0$$

Then, since $\log(1+x) \rightarrow x$ as $x \rightarrow 0$, if follows that, as $s \rightarrow 0$,

$$b_k^* = \log(\eta_k/\theta_k) \rightarrow s\left[f(a_{k-1}) - f(a_k)\right]/\theta_k \triangleq \bar{b}_k \tag{26}$$

On substituting $\bar{b}_k$ into Eq. (15) and using the fact that

$$\sum_{k=1}^{m} f(a_{k-1}) - f(a_k) = f(a_0) - f(a_m) = f(-\infty) - f(\infty) = 0 \tag{27}$$

we obtain the efficacy of T_n^*:

$$\epsilon(T_n^*) = \sum_{k=1}^{m} \left[f(a_{k-1}) - f(a_k)\right]^2/\theta_k \tag{28}$$

It can be shown [9] that

$$\epsilon(T_n^*) \geq \epsilon(T_n) \tag{29}$$

where (T_n^*) is given by Eq. (28) and (T_n) by Eq. (15). This detector can be applied under any noise environment since it was derived without any restriction on the noise distribution. In particular, when the noise density is difficult to formulate analytically, this detector may be desirable. If the actual noise density function is available, the parameters θ_k can be obtained easily and the parameters η_k can be calculated only if the signal is known.

IV. THE LIKELIHOOD RATIO DETECTOR BASED ON THE MULTINOMIAL VECTOR $(N_1, N_2, \ldots, N_m)$ - $\{\eta_k\}$ UNAVAILABLE

In this section, we employ all the assumptions made at the beginning of Section III except assumption (v). Since the η_k are not available, the detection is equivalent to that of testing a simple hypothesis versus a composite alternative:

$$H: \underline{p} = \underline{\theta} \qquad \text{vs.} \qquad K: \underline{p} \neq \underline{\theta}$$

upon observing a multinomial vector $(N_1, N_2, \ldots, N_m)$ with parameters $\underline{p}$ and n, the number of observations. The parameter vector $\underline{p} = (p_1, p_2, \ldots, p_m)$ can range over the following set:

$$\Omega = \left\{\underline{p} \,\middle|\, p_k \geq 0, \quad 1 \leq k \leq m, \quad \sum_{k=1}^{m} p_k = 1\right\} \tag{30}$$

It is known [6] that a UMP test does not exist for such a hypothesis testing problem. We proceed to derive a likelihood ratio detection scheme for our problem. To form the likelihood ratio, we need an estimate $\hat{\underline{p}}$ of $\underline{p}$. Since the a priori distribution of the $\{\eta_k\}$ is not available, the maximum likelihood estimate (MLE) is the most desirable since it maximizes a posteriori probability when a priori information does not introduce a prejudice [4,5]. It can be shown [9] that the MLE of $\underline{p}$ is given by

$$\hat{\underline{p}} = (\hat{p}_1, \hat{p}_2, \ldots \hat{p}_m) = \left(\frac{N_1}{n}, \frac{N_2}{n}, \ldots \frac{N_m}{n}\right) \qquad (31)$$

The likehood ratio can thus be formed:

$$\log L(N_1, \ldots N_m) = \sum_{k=1}^{m} N_k \log (p_k/\theta_k) = \sum_{k=1}^{m} N_k \log (N_k/n\,\theta_k) \qquad (32)$$

The level-α likelihood ratio test scheme is then the following:

$$\log L = \sum_{k=1}^{m} N_k \log (N_k/n\theta_k) \begin{Bmatrix} > \\ < \end{Bmatrix} T(\alpha) \begin{matrix} \Rightarrow K \\ \Rightarrow H \end{matrix} \qquad (33)$$

where $T(\alpha)$ is the test threshold chosen so as to achieve level α.

It should be noted that this detector is not included in the class of detectors defined in Fig. 2 since the weighting coefficients $\log(N_k/n\theta_k)$ involve the PC's output, $(N_1, N_2, \ldots, N_m)$. However, this detector is essentially an adaptive one since it can deal with any signal environment. The information from the signal is characterized by the MLE of the multinomial parameter vector $\hat{\underline{p}}$ given in Eq. (31).

Unfortunately, the probability distribution of (log L) is unmanageable; even the Gaussian approximation technique (the Central Limit Theorem) is not useful in this case. Also, the mean and the variance of log L are difficult to obtain if not impossible. As a matter of fact, (2 log L) is not asymptotically Gaussian. Instead, it has approximately a chi-square distribution with (m-1) degrees of freedom in large samples. This will be shown heuristically in the following. A test statistic called Pearson's χ^2 is defined as

$$\chi^2 = \sum_{k=1}^{m} (N_k - n\theta_k)^2 / n\theta_k \qquad (34)$$

Clearly, χ^2 is a measure of the departure of the multinomial vector $(N_1, N_2, \ldots, N_m)$ from its expectation $(n\theta_1, n\theta_2, \ldots n\theta_m)$ under H. It can be shown [8] that, under H, χ^2 tends in law to a chi-square distribution with (m-1) degrees of freedom as n is large enough so that

$$n\theta_k > 5 \qquad \text{for all } 1 \leq k \leq m \qquad (35)$$

Also, we can prove [9] that

$$\chi^2 \simeq 2 \log L \qquad (36)$$

under H when n is large. Therefore, we can approximate the probability distribution of (2 log L) by the chi-square distribution with (m-1) degrees of freedom under H in large samples. It immediately follows that, approximately,

$$T(\alpha) \approx \frac{1}{2} \chi^{-1}_{m-1} (1-\alpha) \qquad (37)$$

if sample size n satisfies the condition (36), where $\chi^{-1}_{m-1}(u)$ denotes the u-th quantile of the chi-square distribution with(m-1) degrees of freedom.

In practice, we can use χ^2 as the test statistic of our detector instead of (log L). In this case, the test threshold should be $2T(\alpha)$, of course. It was shown [5] that the χ^2 test is consistent (and is thus asymptotically unbiased).

The same kind of approximation procedure can be applied to find the power function. If the signal "s" is so small that, under K, $\underline{p}(s)$ is close to $\underline{\boldsymbol{\theta}}$, then χ^2 can be treated as having a noncentral chi-square distribution with (m-1) degrees of freedom when K is true (see [5]). The noncentrality parameter is given by

$$\delta = n \sum_{k=1}^{m} \left[p_k(s) - \theta_k\right]^2 / \theta_k \tag{38}$$

Denote this chi-square distribution by $G_s(\cdot)$. The power function can be expressed as

$$\beta(s) = 1 - G_s\left[\chi^{-1}_{m-1}(1-\alpha)\right] \tag{39}$$

If the functional form of the actual noise density is known, $p_k(s)$ can be calculated. In this case, $G_s(\cdot)$ is in an explicit form and the power function can be plotted easilty.

V. COMMENTS ON DESIGNING A PC

The nonparametric property of the detectors discussed above is due to the PC device. The PC can be defined by a set of real parameters $(a_1, a_2, \ldots, a_{m-1})$, as can be seen in Fig. 1. Certainly, the smaller is the integer m the simpler is the structure of the PC. Given an m, however, there is no optimal choice of $(a_1, a_2, \ldots a_{m-1})$ that can best retain information from the input data for an arbitrary statistical characteristic. In the following, we discuss some ways for selecting these PC parameters.

A. Selection of $(a_1, a_2, \ldots, a_{m-1})$ According to the Noise Distribution

In [1], Ching and Kurz proposed an "m-interval partitioning". Such a partition requires an estimate of the(m-1) quantiles of the noise distribution, $\{F^{-1}(1/m), F^{-1}(2/m), \ldots, F^{-1}[(m-1)/m]\}$. On the other hand, we do not need to estimate those multinomial parameters under the hypothesis since each of them is equal to (1/m). Although there is no evident reason to support using a "uniform" partitioning like this, a PC structure thus developed seems to be convenient and reasonable. Two disadvantages associated with this approach are: (i) the PC parameters vary with the noise environment, and (ii) a good estimate of quantiles is often difficult to obtain.

B. Selection of $(a_1, \ldots, a_{m-1})$ when the Signal is Known Exactly

For the special case where the exact signal strength is available, it is reasonable to pick the a_i's so that

$$a_i - a_{i-1} = s \,, \quad i=2,1,..m-1 \tag{40}$$

where s is the signal strength. Then the multinomial parameters under the hypothesis and under the alternative have the following relationship:

$$\eta_i = \theta_{i-1} \;,\; i=3,4,..m-1 \qquad \text{and} \qquad \eta_m = \theta_{m-1}+\theta_m \;,\; \eta_1+\eta_2=\theta_1 \tag{41}$$

Therefore, the detector can be implemented on estimating m real parameters.

C. Selection of $(a_1,a_2,...,a_{m-1})$ According to the Noise Variance

Suppose the noise variance σ^2 is available when we design the PC. A convenient partitioning is now the following:

$$a_i = -\sigma+(i-1)\frac{2\sigma}{m-2} \;,\quad i=1,2,...,m-1 \tag{42}$$

If we always normalize the input data so that $\sigma^2=1$, then we have

$$a_i = -1+ \frac{2(i-1)}{m-2} \;,\qquad i=1,2,...m-1 \tag{43}$$

VI. PERFORMANCES

In Section III, we assumed that the PC device is well designed and a Neyman-Pearson optimal detection scheme was derived based on observing this PC's output. We now investigate how such a detector performs when the accompanying PC is designed according to the strategies we mentioned in Section V.

For convenience, a PC device will be called PC I if its structure is what we discussed in Section V.(A). Similarly, the names PC II and PC III are given to those mention in Section V.(B) and V.(C) respectively. For each case, the test power of the corresponding Neyman-Pearson optimal detector is computed for Gaussian noise for a fixed test level. When these results are compared to the power of the parametric Neyman-Pearson optimal detector as shown in Table 1, these nonparametric detectors demonstrate their usefulness.

		β		
m	N-P for PC I	N-P for PC II	N-P for PC III	parametric N-P optimal
4	0.9939	0.9802	0.9947	0.9976
5	0.9952	0.9829	0.9957	0.9976
6	0.9959	0.9851	0.9960	0.9976
7	0.9963	0.9870	0.9961	0.9976
8	0.9966	0.9899	0.9962	0.9976
9	0.9968	0.9910	0.9963	0.9976
10	0.9969	0.9919	0.9963	0.9976

Table 1. Comparison between the N-P optimal detector discussed in Section III and the parametric N-P optimal detector in terms of the power β of the test. Gaussian noise, $n=2000$, $\alpha=0.05$, $s=0.1$

When the signal strength is unknown the likelihood ratio detector proposed in Section IV applies. For PC I and PC III, the power function of the corresponding likelihood ratio detector is plotted in Fig. 3 for Gaussian noise using Eqs. (38) and (39) and the appropriate IMSL library functions. It can be seen that PC I and PC III are approximately as efficient in this case. Also shown in Fig. 3 for comparsion is the power function for the parametric level-α test which is obtained upon observing the input data directly.

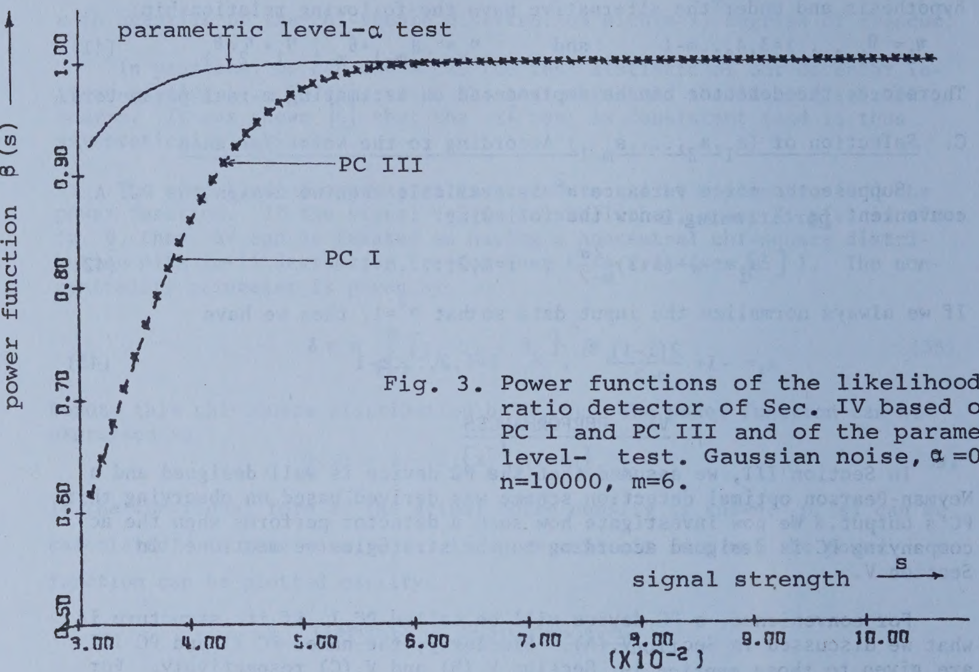


Fig. 3. Power functions of the likelihood ratio detector of Sec. IV based on PC I and PC III and of the parametric level- test. Gaussian noise, α =0.05, n=10000, m=6.

REFERENCES

1. Y.C. Ching and L. Kurz, "Nonparametric Detectors Based on m-Interval Partitioning:, IEEE Trans. Information Theory, Vol. IT-18, No.2, March, 1972, pp. 251-257.
2. J.B. Thomas, "Nonparametric Detection", Proc. of the IEEE, Vol. 58, No. 5, May, 1970, pp. 623-631.
3. S.A. Kassam and J.B. Thomas,"Generalizations of the Sign Detector Based on Conditional Tests", IEEE Trans. Comm., Vol. Com-24, No.5, May, 1976, pp.481-487.
4. C.W. Helstrom, Statistical Theory of Signal Detection, Pergamon Pr., New York, 1968.
5. M.G. Kendall and A. Stuart, The Advanced Theory of Statistics, Vol. 2, Hafner, New York, 1961.
6. P.J. Bickel and K.A. Doksum, Mathematical Statistics:Basic Ideas and Selected Topics, Holden-Day, San Francisco, 1977.
7. I. Selin, Detection Theory, Princeton Univ. Pr., Princeton, NJ, 1965.
8. H. Cramer, Mathermatical Methods of Statistics, Princeton Univ. Pr., Princeton, NJ, 1946.
9. C.C. Lee, On Nonparametric, Sequential and Mixed-Sample-Size Signal Detection Procedures, Ph.D. Dissertation, Dept. of EECS, Princeton University, Princeton, New Jersey, July, 1980.

ACKNOWLEDGEMENTS

The authors acknowledge the support of the Air Force Office of Scientific Research under Grant AFOSR-76-3083.

THE PERFORMANCE CHARACTERISTICS OF TWO EXTENSIONS OF THE SIGN DETECTOR

SALEEM A. KASSAM
Department of Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

Two related techniques have been proposed in the past for improving the performance of the sign detector, through higher-order data quantization. The fixed-threshold m-interval detector and the generalized sign detector using a conditional test are both nonparametric detectors which are fairly simple to implement. In this paper we compare the asymptotic and finite-sample, finite-signal performance characteristics of these two detectors, and point out their relative advantages and disadvantages.

I. INTRODUCTION

For the detection of a deterministic signal in additive noise with zero median, the sign detector is an easily implemented nonparametric detector with a constant probability of type I error (false alarm). Let $\underline{X} = (X_1, X_2, \ldots, X_n)$ be the vector of possibly signal-bearing observations, so that

$$X_i = W_i + \theta s_i, \quad i=1,2,\ldots,n, \tag{1}$$

where $\theta \geq 0$ and the W_i are independent random variables representing noise. We will denote by F and f the common probability distribution and density functions, respectively, of the W_i. The signal components s_i are assumed to be non-zero.

In the constant signal case the sign detector test statistic for testing $\theta=0$ vs. $\theta>0$ is

$$T = \sum_{i=1}^{n} \operatorname{sgn}(X_i) \tag{2}$$

where

$$\operatorname{sgn}(x) = \begin{cases} 1, & x>0 \\ -1, & x\leq 0 \end{cases}$$

For a time-varying signal, the above statistic can be modified to

$$\tilde{T} = \sum_{i=1}^{n} \operatorname{sgn}(s_i)\operatorname{sgn}(X_i) \tag{3}$$

or to

$$\hat{T} = \sum_{i=1}^{n} s_i \operatorname{sgn}(X_i) \tag{4}$$

The statistic $\tilde{T}$ is simpler to implement than $\hat{T}$, but has a somewhat lower efficiency, as will be seen later.

The performance of the sign detector is rather poor for noise densities such as the Gaussian, when compared to the corresponding optimal (parametric) detectors and to more complex nonparametric detectors based on data ranking. In order to obtain improved performance over that of the sign detector, without a major increase in complexity, two schemes have been proposed based on a higher-order partitioning of the observations. The m-interval detector has been described in [1], and the generalized sign detector was first formulated in [2]. In this paper we will compare the relative performances and characteristics of these two detectors. We will consider the asymptotic

---

This research is supported by the Air Force Office of Scientific Research under Grant AFOSR 77-3154.

performance of the detectors (for $\theta \to 0$ and $n \to \infty$), as well as finite-sample characteristics. Both of these extensions of the simple sign detector have also been applied in schemes for the detection of random signals, in multi-input systems, for sequential detection, and in other applications ([3] - [6]). However, the comparisions we will present for the deterministic signal case with a single fixed-length observation vector will bring out the main features of these two schemes.

In the next section we describe briefly the m-interval and generalized sign detectors, before considering their performance characteristics in Section III.

II. EXTENSIONS OF THE SIGN DETECTOR

Let us first assume that the signal in (1) is constant, so the $s_i = s > 0$, $i=1,2,\ldots,n$. Consider a vector $\underline{a} = (a_0, a_1, \ldots, a_m)$, where $a_0 = -\infty$, $a_m = \infty$ and $a_j > a_{j-1}$, $j=1,2,\ldots,m$. We define intervals

$$I_j = (a_{j-1}, a_j], \quad j=1,2,\ldots,m \tag{5}$$

and denote by N_j the number of observations X_i falling in I_j. Now suppose that the a_j are quantiles of the noise distribution F, so that $F(a_j) = j/m$. In this case the test statistic defined by

$$Q = \sum_{j=1}^{m} b_j N_j, \tag{6}$$

where $\underline{b} = (b_1, b_2, \ldots, b_m)$ is a deterministic weight vector, can be used to form a nonparametric test for $\theta=0$ vs. $\theta>0$ in (1). This is because for any noise distribution function F with quantiles given by $\underline{a}$, the distribution of Q when $\theta=0$ is exactly known. A detector formed according to this scheme is known as an m-interval detector [1]. Note that the sign detector is obtained when $m=2$, $a_1=0$ and $b_1=-1$, $b_2=1$. It is also clear that $\underline{a}$ may be any vector of partitioning parameters a_j for which the $F(a_j) = p_j$ are known. It may be more reasonable to assume that the $\underline{a}$ vector corresponding to the values $p_j = j/m$ can be estimated from prior data. The weight vector $\underline{b}$ would normally be picked to result in a good compromise in performance over some collection of possible density functions f.

Let us now assume that the noise density function is symmetric, so that $f(x) = f(-x)$. Let the vector $\underline{a}$ have odd-symmetry, that is, $a_{m-\ell} = -a_\ell$, $\ell=1, 2,\ldots,m-1$. If f is symmetric, an odd-symmetric $\underline{a}$ would be required for the m-interval detector if $p_j = j/m$. In addition, as will be seen in the next section, the optimum choice of $\underline{a}$ for given symmetric f is odd-symmetric.

The generalized sign detector [2] based on any odd-symmetric $\underline{a}$ vector achieves a fixed type I error probability for any symmetric density function of the noise. Its test statistic is also defined by (6), but the test is formed by a comparision with a variable threshold, with a variable randomization probability when Q is equal to the threshold. Note that knowledge only of the symmetry of f is not sufficient to yield the distribution of Q when $\theta=0$. Consider, however, the statistic $\underline{C} = (C_1, C_2, \ldots, C_p)$ with

$$C_\ell = N_\ell + N_{m-\ell+1}, \quad \ell=1,2,\ldots,p, \tag{7}$$

with $p=[m/2]$, the largest integer less than m/2. Now conditioned on $\underline{C}$, the distribution of Q when $\theta=0$ is completely defined; the distribution of N_ℓ given C_ℓ is binomial, independent of the binomial distribution of N_j given C_j for $j \neq \ell$. Thus a nonparametric test can be implemented with thresholds and randomization probabilities computed as functions of $\underline{C}$. As an example, for a four-level test ($m=4$) the test parameters are functions of the scalar $C_1 = N_1 + N_4$. It should be noted that here also the case $m=2$ reduces to the sign test. This detector structure is shown in Fig. 1.

The m-interval detector is based on the assumption that the vector of quantiles $\underline{a}$ for the noise is known, or at least that the $F(a_j)$ values are known for the components of a given $\underline{a}$, for the class of allowable noise densities. Such classes of noise densities are nonparametric classes, but are somewhat restrictive in their membership. For example, for m=3 or 4 with an odd-symmetric $\underline{a}$ vector of quantiles, only one Gaussian density can belong to the allowable class. For m=5 and some assumed quantile vector $\underline{a}$, no Gaussian density (or any other parametric density with only one or two free parameters) may belong to the allowable class. The generalized sign detector needs a conditional structure which represents a slight increase in implementation cost, but gives nonparametric performance for the large class of symmetric densities for the noise.

Since the m-interval detector is based on more specific assumptions about the class of allowable noise densities, its detection performance can be expected to be better than that of the conditional-test implementation leading to the generalized sign detector. It will be found, however, that the difference in performance is small for most cases of interest. Before we take up the performance comparisions, we will briefly indicate how both detectors above may be modified for the case where the s_i, i=1,2,...n, are not all equal.

Note that the test statistic Q of (6) can be written as [7]

$$Q = \sum_{j=1}^{m} b_j \sum_{i=1}^{n} Z_{ij}, \qquad (8)$$

where

$$Z_{ij} = \begin{cases} 1, & X_i \varepsilon I_j \\ 0, & \text{otherwise} \end{cases}, \qquad (9)$$

so that $N_j = \sum_{i=1}^{n} Z_{ij}$. Simple modifications of Q which lead to more efficient tests for non-constant signals may be defined as the test statistics

$$\tilde{Q} = \sum_{j=1}^{m} b_j \sum_{i=1}^{n} \operatorname{sgn}(s_i) Z_{ij} \qquad (10)$$

and

$$\hat{Q} = \sum_{j=1}^{m} b_j \sum_{i=1}^{n} s_i Z_{ij} \qquad (11)$$

For the m-interval detector, it is assumed that for $\theta=0$ the matrix $\underline{Z}$ of random variables Z_{ij} has a known distribution. For the generalized sign detector, based on the symmetry of the noise density functions, we also have known distribution for $\underline{Z}$ conditioned on $\underline{C}$ of (7). Thus both $\tilde{Q}$ and $\hat{Q}$ can be used for a conditional-test generalized sign detector. It is apparent that $\tilde{Q}$ and $\hat{Q}$ reduce to $\tilde{T}$ and $\hat{T}$, respectively, when m=2 and $b_2 = -b_1 = 1$.

III. PERFORMANCE CHARACTERISTICS

(a) Asymptotic Performance

The asymptotic relative efficiency (ARE) of two detectors is the ratio of sample-sizes required by the detectors to maintain the same performance (error probabilities) in the limiting case when signal strength approaches zero, and thus sample sizes tend to infinity. In many cases the ARE of two detectors is simply the ratio of their efficacies [8], the efficacy ε of a test based on a statistic S being defined by

$$\varepsilon = \lim_{n\to\infty} \frac{\left[\frac{d}{d\theta} E\{S\}\Big|_{\theta=0}\right]^2}{n \operatorname{Var}\{S\}\Big|_{\theta=0}} \tag{12}$$

Thus ε is like a signal-to-noise ratio measure for weak signals.

An interesting conclusion may be drawn from the above. This is that the ARE of the m-interval detector relative to the generalized sign detector is unity, when $\underline{a}$, $\underline{b}$, are the same for both detectors, based on any one of the three test statistics of (8), (10), or (11). Although this follows from the fact that the efficacies are then the same for both detectors, some care is needed in proving that this implies an ARE of unity, since the generalized sign detector uses a conditional test. This type of proof has been outlined in [4].

This result implies that for large sample sizes there will be very little difference in performance between the two detectors, even though one is based on a more specific set of assumptions about the symmetric noise density function. This may be intuitively explained by the fact that, in principle, for large sample sizes good estimates of the noise quantiles may be obtained from observed data (signal-bearing or noise-only), given that the noise density is symmetric.

From the definition of (12), it can be shown directly that $\hat{\varepsilon}$, the efficacy of the detectors based on $\hat{Q}$ of (10), is given by

$$\hat{\varepsilon} = \overline{s^2}\, \frac{\left\{\sum_{j=1}^{m} b_j [f(a_{j-1}) - f(a_j)]\right\}^2}{\sum_{j=1}^{m} b_j^2 [F(a_j) - F(a_{j-1})] - \left\{\sum_{j=1}^{m} b_j [F(a_j) - F(a_{j-1})]\right\}^2} \tag{13}$$

where $\overline{s^2} = \lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} s_i^2$. For $\tilde{Q}$, the efficacy $\tilde{\varepsilon}$ is given by (13) with $\overline{s^2}$ replaced by $\overline{|s|}^2 = \lim_{n\to\infty} [\frac{1}{n} \sum_{i=1}^{n} |s_i|]^2$. Note that $\overline{|s|}^2 \le \overline{s^2}$. It is easily shown that for given $\underline{a}$, the best choice of $\underline{b}$ maximizing $\tilde{\varepsilon}$ or $\hat{\varepsilon}$ is

$$b_j = K \frac{f(a_{j-1}) - f(a_j)}{F(a_j) - F(a_{j-1})}, \quad j=1,2,\ldots,m, \tag{14}$$

where K>0 is any constant. The optimum $\underline{a}$ vector for some assumed noise density f can then also be obtained; this $\underline{a}$ is odd-symmetric for symmetric f. The vector of quantiles is not necessarily the best choice for $\underline{a}$ for given f. However, the quantiles may be reasonable if the detector is to be implemented as a nonparametric detector for the class of allowable noise densities whose quantiles are assumed known based on prior measurements.

(b) Finite-Sample Performance

One of the interesting questions which we now consider is the degree of performance degradation in the generalized sign detector as compared to the m-interval detector, for finite sample sizes and finite signals. The numerical performance results will be obtained for the case where m=4. Although results for higher-order partitioning can be obtained in principle with no additional programming difficulties, the resulting computation time requirements become rather large. The case where m=3 or 4 usually represents a reasonable compromise between improved performance and simple implementation. In any case, the general characteristics of the relative performances of the two detectors should remain the same for larger m. We will also focus on the constant signal case, although again the time-varying signal case can be analysed with more computation time and should yield similar characteristics.

In obtaining numerical results for detection power, the parameters $\underline{a}$ and $\underline{b}$ for the detectors were chosen to yield a good compromise design (based on the analytical efficacy expressions) for the class of noise density functions containing the Gaussian, double-exponential and Cauchy densities. These last two densities are often used in judging the performance of detectors under conditions of heavy-tailed non-Gaussian noise.

For the symmetric densities and odd-symmetric $\underline{a}$ vector, we will have $a_0 = -\infty$, $a_2 = 0$, and $a_4 = \infty$, for the case m=4. For zero-mean Gaussian noise density with variance σ^2, the quantiles a_1 and a_3 are given by $a_3 = -a_1 = 0.6745\sigma$. We will now assume that for the m-interval detector the known (or estimated) quantiles of the symmetric noise distribution are defined by the components of the above $\underline{a}$ vector with $a_3 = -a_1 = 0.6745\sigma_0$ where σ_0 is a known positive constant. This definition of the parameter $\underline{a}$ may be thought of as arising from an assumption that the noise density function is nominally Gaussian with variance $\sigma_0{}^2$.

The double-exponential density function is defined by

$$f_e(x) = \frac{1}{\sigma\sqrt{2}} e^{-|x|\sqrt{2}/\sigma} \qquad (15)$$

where σ^2 is the variance. In this case the quantiles a_1 and a_3 are given by $a_3 = -a_1 = 0.4901\sigma$. Thus $f_e(x)$ with $\sigma = 1.376\sigma_0$ has the same quantiles as the Gaussian density with variance σ_0^2 for m=4.

For the Cauchy density function

$$f_c(x) = \frac{1}{\pi k} \frac{1}{1 + (x/k)^2} \qquad (16)$$

we have $a_3 = -a_1 = k$; thus $k = 0.6745\sigma_0$ gives the same quantiles (for m=4) as the Gaussian density with variance $\sigma_0{}^2$.

For these symmetric densities, we find from (14) that the optimum $\underline{b}$ vector has odd symmetry, so that $b_4 = -b_1$ and $b_3 = -b_2$. Proceeding to find these asymptotically optimum values as suggested in (a) above, we find that the optimum value of $\underline{b}$ is defined by $b_3 = 0.2554b_4$ for Gaussian noise with variance $\sigma_0{}^2$, and by $b_4 = b_3$ for both the density f_e with $\sigma=1.376\sigma_0$ and the density f_c with $k=0.6745\sigma_0$. In all cases the vector $\underline{a}$ was the common vector of quantiles, m=4. Note that this says that the sign detector is optimal for the Cauchy density if the partitioning is based on the quantiles for m=4. The sign detector is also the maximum-efficacy (locally optimum) detector for double-exponential noise.

As a compromise design for the weight-vector $\underline{b}$, we will use $b_3=0.5b_4$. Together with the quantile vector $\underline{a}$, this leads to an efficacy of $0.82/\sigma_0{}^2$ for the Gaussian density with variance $\sigma_0{}^2$, an efficacy of $0.95/\sigma_0{}^2$ for the double-exponential density with the same quantiles, and an efficacy of $0.80/\sigma_0{}^2$ for the Cauchy density wiht the same quantiles. Note that with the respective optimum analog schemes, the maximum efficacies in these three cases are [7] $1/\sigma_0{}^2$, and $1.06/\sigma_0{}^2$ and $1.1/\sigma_0{}^2$. Using a sign detector on these fixed-quantile densities, the efficacies are $0.64/\sigma_0{}^2$, $1.06/\sigma_0{}^2$, and $0.89/\sigma_0{}^2$, respectively. On the other hand, using the linear sum detector which gives the maximum efficacy for Gaussian noise, we get an efficacy of $0.53/\sigma_0{}^2$ and zero for f_e and f_c, respectively, with the same quantiles. Thus, the compromise design where $\underline{a}$ is the quantile-vector and $\underline{b}$ is defined by $b_3 = 0.5b_4$ is seen to yield a useful design with good overall asymptotic performance compared to the sign detector and the linear detector.

For the vectors $\underline{a} = (-\infty, -0.6745\sigma_0, 0, 0.6745\sigma_0, \infty)$ and $\underline{b} = (-2, -1, 1, 2)$, numerical results on detection powers were obtained for sample sizes $n=25$ and 50, false-alarm probabilities $\alpha=10^{-2}$ and 10^{-3} and for a range of values of θ/σ_0 [for constant-signal detection so that $s_i=1$, $i=1, \ldots n$, in (1)] for both the four-interval detector and the four-level generalized sign detector. These results are given in Table I. In this table, detection probabilities are given for the three noise density functions we have discussed previously, each density function having the common quantiles defined above. For each density function detection probabilities are given for the fixed-threshold m-interval test and the conditional generalized sign test.

Several interesting features are apparent from Table I. As expected, we observe that the conditional test has a power which is smaller than that of the fixed-threshold test. It can be seen that for small values of θ/σ_0, a condition under which the efficacy of a detector is a good indication of its detection power (for n not too small), the conditional and fixed-threshold test powers are almost the same. We also find that in this case the variation in detection power between the three different noise density functions is in agreement with the variation in efficacy for these cases. As θ/σ_0 increases, there is a more apparent difference between the powers of the fixed-threshold and conditional tests. However, for the Gaussian case this difference in powers is quite nominal. Compared to the performance of the sign detector, both the conditional and fixed-threshold tests achieve the same high degree of improvement for Gaussian noise.

The double-exponential and Cauchy noise densities seem to lead to somewhat larger differences in powers for the two tests. This may be due to the possibility that knowledge of the quantiles (m-interval fixed-threshold test implementation) amounts to having more "information", in the detection context, for the broad-tailed non-Gaussian densities as compared to the Gaussian case, so that larger sample sizes are needed before the conditional and fixed-threshold tests have similar performances.

The detection powers in Table I also reveal that for small signal strength the sign detector performs better for the double-exponential density than the four-level detectors. However, it is interesting that this performance advantage disappears for larger values of θ/σ_o. This is due to the fact that although the sign detector is locally optimum for double-exponential noise, for non-vanishing θ the optimum detector nonlinearity is not the hard limiter, but the "soft" limiter function $f_e(x-\theta)/f_e(x)$. Thus for larger values of θ the four-level quantizer characteristics of the four-interval fixed-threshold and conditional-threshold detectors give a closer match to the "soft" limiter [7]. For Gaussian noise the four-level detectors are uniformly better than the sign detector, because the uniformly most powerful linear-sum detector is always better approximated by the four-level quantizer characteristic. For the Cauchy density, the variations in relative performance of the four-level and sign detectors are again explained by similar considerations.

Although the fixed-threshold four-interval detector achieves a larger detection power than the conditional threshold detector, it achieves a fixed false-alarm probability provided the noise quantile vector is correctly specified. If the actual noise density function has a quantile vector other than the assumed vector, the false alarm probability α of the fixed threshold four-level detector will differ from the design value α_0. The conditional-threshold implementation, on the other hand, retains the design value of α for any symmetric noise density function.

Table II shows the effect on false alarm probabilities of a mis-match between the design quantile vector a and the actual values of the quantile vector for the noise density function in the fixed-threshold detector. The first four columns of the table show the actual values of $p_j = F(a_j)$, where F is the noise distribution function. The results are shown for two different design values α_0 of the false alarm probability, achieved when $p_j = 0.25$, $j = 1,2,3,4$. It is seen that the actual value of α may differ considerably from the design value, especially for small design values of α. These results also show that in order to ensure a value for α which tends to be less than or equal to a value α_0, the implemented value of the parameter $a_3 = -a_1$ in a should be taken to be somewhat larger than the nominal or estimated value. This would result in a lower detector power when the nominal assumptions are correct. Table III illustrates this effect for the case of Gaussian noise with variance σ_0^2, where the four-level detection is based on a value $a_3 = 0.82\sigma_0$ rather than the nominal value $a_3 = 0.6745\sigma_0$. In this case it is seen that the conditional test performs almost as well as, or better than, the fixed threshold detector. In fact, the power of the conditional test in Table III is better than in the corresponding situation in Table I. This is because the optimum value for a_3 here is $0.98\sigma_0$ [7].

The fundamental difference between the m-interval (fixed-threshold) and generalized sign (conditional test) detectors is that in the latter, the threshold and randomization probability depend on the received data. The conditional test is, nevertheless, fairly easy to implement; one method of implementation is to use a stored set of thresholds and randomizations in a table look-up device, as described in [2]. One simplification which may be desireable in any threshold test is the quantization of the randomization probability, to perhaps only two values zero and 0.5. This would also reduce storage requirements in the table for a conditional test, only one bit now being required for each randomization probability.

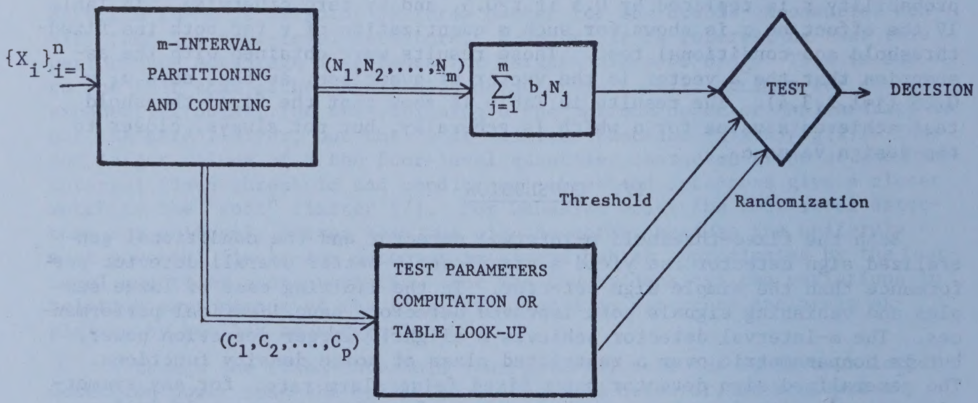
In nonparametric applications the design value of the false alarm probability α_0 would not be exceeded if in each case the exact randomization probability r is replaced by 0.5 if $r \geq 0.5$, and by zero otherwise. In Table IV the effect on α is shown for such a quantization of r for both the fixed-threshold and conditional test. These results were obtained with the assumption that the a vector is the vector of quantiles, so that each $p_j = 0.25$ $(j=1,2,3,4)$. The results in Table IV show that the fixed-threshold test achieves a value for α which is generally, but not always, closer to the design value α_0.

IV. CONSLUSION

Both the fixed-threshold m-interval detector and the conditional generalized sign detector can yield a signficantly better overall detector performance than the simple sign detector. In the limiting case of large samples and vanishing signals both improved detectors have identical performances. The m-interval detector achieves a slightly larger detection power, but is nonparametric over a restricted class of noise density functions. The generalized sign detector has a fixed false alarm rate for any symmetric noise density function, at the expense of a minor increase in implementation complexity. If the quantile vector in the m-interval detector is imprecisely known, nonparametric performance is possible only with a loss in power. For the generalized sign detector, knowledge of the quantiles is not essential, but serves as a guide in picking a good partitioning scheme for the data.

REFERENCES

1. Y.C. Ching and L. Kurz, "Nonparametric detection based on m-interval partitioning," IEEE Trans. Inform. Theory, Vol. IT-17, pp. 251-257, March 1972.
2. S.A. Kassam and J.B. Thomas, "Generalizations of the sign detector based on conditional tests," IEEE Trans. Commun., vol. COM-24, pp. 481-487, May 1976.
3. L. Kurz, "Nonparametric detectors based on partition tests," in Nonparametric Methods in Communications, P. Papantoni-Kazakos and D. Kazakos, Eds; Marcel Dekker, New York, 1977, pp. 73-143.
4. S.A. Kassam, "Conditional tests in nonparametric detection," in Nonparametric Methods in Communications, P. Papantoni-Kazakos and D. Kazakos, Eds; Marcel Dekker, New York, 1977, pp. 145-201.
5. J.G. Shin and S.A. Kassam, "Multilevel coincidence correlators for random signal detection," IEEE Trans. Inform. Theory, vol. IT-25, pp. 47-53, January 1979.
6. P. Kersten and L. Kurz, "Bivariate m-interval classifers with application to edge detection," Inform. Contr., vol. 34, pp. 152-168, June 1977.
7. S.A. Kassam, "Optimum quantization for signal detection," IEEE Trans. Commun., vol. COM-25, pp. 479-484, May 1977.
8. J. Capon, "On the asymptotic efficiency of locally optimum detectors," IRE Trans. Inform. Theory, vol IT-7, pp. 67-71, April 1961.



$\{X_i\}_{i=1}^n$
m-INTERVAL PARTITIONING AND COUNTING
$(N_1, N_2, \ldots, N_m)$
$\sum_{j=1}^{m} b_j N_j$
TEST
DECISION
Threshold
Randomization
TEST PARAMETERS COMPUTATION OR TABLE LOOK-UP
$(C_1, C_2, \ldots, C_p)$

FIGURE 1 Structure Of Generalized Sign (Conditional) Detector

TABLE I DETECTION POWER OF m-INTERVAL (FIXED THRESHOLD) AND GENERALIZED SIGN (CONDITIONAL) DETECTORS , m=4

	θ/σ_0	Gaussian		Double Exponential		Cauchy	
		Fixed Threshold	Conditional	Fixed Threshold	Conditional	Fixed Threshold	Conditional
N = 25 α = 0.01 m = 4	0	0.010	0.010	0.010	0.010	0.010	0.010
	0.05	0.0178	0.0175	0.0183	0.0183	0.0176	0.0176
	0.25	0.1133	0.1077	0.1143	0.1104	0.1091	0.1046
	0.50	0.4589	0.4341	0.4296	0.3967	0.4173	0.3758
	0.75	0.8387	0.8129	0.7895	0.7333	0.7531	0.6796
	1.00	0.9805	0.9727	0.9512	0.9183	0.9208	0.8631
N = 50 α = 0.001 m = 4	0	0.001	0.001	0.001	0.001	0.001	0.001
	0.05	0.0028	0.0027	0.0029	0.0029	0.0027	0.0027
	0.25	0.0675	0.0613	0.0665	0.0633	0.0616	0.0578
	0.50	0.5230	0.4931	0.4793	0.4366	0.4603	0.4055
	0.75	0.9458	0.9313	0.9088	0.8646	0.8765	0.8094
	1.00	0.9991	0.9985	0.9944	0.9861	0.9854	0.9616
N = 50 α = 0.001 m = 2 (Sign Detector)	0	0.001		0.001		0.001	
	0.05	0.0024		0.0030		0.0029	
	0.25	0.0429		0.0652		0.0647	
	0.50	0.3545		0,4092		0.4220	
	0.75	0.8266		0.8051		0.7965	
	1.00	0.9869		0.9650		0.9474	

TABLE II ACTUAL TYPE I ERROR PROBABILITIES α OF m-INTERVAL DETECTOR, m=4, FOR DESIGN VALUES α_0

Parameters of actual noise distribution				N = 25		N = 50	
p_1	p_2	p_3	p_4	$\alpha_0=0.01$	$\alpha_0=0.001$	$\alpha_0=0.01$	$\alpha_0=0.001$
0.20	0.30	0.30	0.20	0.0065	0.0005	0.0065	0.0005
0.225	0.275	0.275	0.225	0.0082	0.0007	0.0082	0.0007
0.25	0.25	0.25	0.25	0.010	0.001	0.010	0.001
0.275	0.225	0.225	0.275	0.0120	0.0014	0.0119	0.0014
0.30	0.20	0.20	0.30	0.0141	0.0018	0.0141	0.0018

TABLE III EFFECT ON DETECTION POWER IN m-INTERVAL DETECTOR, m=4, OF LARGER a_3 PARAMETER TO MAINTAIN α WITHIN DESIGN VALUE

	Gaussian Noise $a_3=0.82\sigma_0$ N = 25 $\alpha_0 = 0.01$	
θ/σ_0	Fixed Threshold	Conditional
0	0.0069	0.010
0.25	0.0909	0.1089
0.50	0.4142	0.4398
0.75	0.8132	0.8196
1.0	0.9763	0.9750

TABLE IV EFFECT ON α OF TWO-LEVEL RANDOMIZATION PROBABILITY

	$\underline{a}$ = quantile vector			
	N = 25		N = 50	
α_0	Fixed Threshold	Conditional	Fixed Threshold	Conditional
0.05	0.0438	0.0442	0.0490	0.0456
0.01	0.0092	0.0085	0.0099	0.0089
0.001	0.00078	0.00073	0.00092	0.00088

ON THE PERFORMANCE OF A MODIFIED SIGN DETECTOR FOR STRONG MIXING NOISE

D. R. HALVERSON
Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843

and

G. L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

It is shown how a modified sign detector may be designed for the nonparametric detection of a constant signal in strong mixing noise. An upper bound on the performance (as measured by the asymptotic relative efficiency) of such detectors is obtained, and the form is specified for a detector whose loss of nonparametric nature is arbitrarily small and whose performance is arbitrarily close to optimal for a large class of noise processes.

I. INTRODUCTION

In situations where knowledge of the noise statistics is sufficiently inexact, it is often desirable to employ a nonparametric detector. If the noise process is independent and identically distributed, a popular choice is the well known sign detector [1]. When dependency is present between samples, however, the classical sign detector must be modified to insure that its nonparametric nature is not compromised. In [2] a method for modifying the sign detector (and other "white noise" nonparametric detectors as well) is presented for the case of the detection of a constant signal in m-dependent noise, and in [3] we extend some of the results of [2]. In this paper we investigate how the approach of [2] and [3] may be employed in the context of the sign detector when the noise is a member of the much wider class of strong mixing processes.

II. PRELIMINARIES

Let $\{X_i;i=1,2,\ldots\}$ be a strictly stationary sequence of random variables. For $a\leq b$, define $M(a,b) = \sigma\{X_a,X_{a+1},\ldots,X_b\}$, the σ-algebra generated by the indicated random variables, where a and b may take on extended real values. Then $\{X_i;i=1,2,\ldots\}$ is strong mixing if there exists a sequence $\{\alpha_i;i=1,2,\ldots\}$ with $\alpha_i\to 0$ such that for each k, $1\leq k<\infty$, and for each $i\geq 1$, $E_1\in M(1,k)$ and $E_2\in M(k+i,\infty)$ together imply $|P(E_1\cap E_2)-P(E_1)P(E_2)|\leq\alpha_i$. Consider now the employment of the classical sign detector for the detection of a constant positive signal in strong mixing noise with number of samples N_0. We will wish to decide between

$$H_0: Y_i = X_i; \quad i=1,\ldots,N_0$$
$$H_1: Y_i = X_i+s; \quad i=1,\ldots,N_0,$$

where we observe realizations $\{y_i;i=1,\ldots,N_0\}$ of the process $\{Y_i;i=1,\ldots,N_0\}$ and s is a positive constant. As a specific example, let $\{X_i;i=1,2,\ldots\}$ be a two state Markov process taking values ±1 with each prior probability equalling 0.5 and where the probability of X_i and X_{i+1} having the same value is 0.525. From [4, p. 167] we infer that this process is strong mixing possessing an α-representation $\{\alpha_i;i=1,2,\ldots\}$ satisfying $\alpha_1 \leq 0.05$ and $\alpha_2 \leq 0.0025$. This is therefore a process which might be judged from an engineering standpoint to be "almost" independent, however, in the context of the sign detector a calculation for $N_0 = 10$ and threshold $T = 8$ shows that the true false alarm probability $\tilde{\alpha}$ exceeds that obtained under an independence assumption $\alpha = 0.01$ by 33%. Contrary to the remarks of [2], a more careful analysis for the case of strong mixing noise is therefore required.

As applied to the sign detector, the method of [2] requires grouping the samples into blocks of length n with m samples skipped between blocks, resulting in $N = G[(N_0+m)/(n+m)]$ independent blocks, where $G(\cdot)$ is the greatest integer function and N_0 is the total number of samples. In our work none of n, m, and N will necessarily be fixed as N_0 increases. If a real-valued transformation is applied to each block, one obtains a sequence $Z_{1,n},\ldots,Z_{N,n}$ upon which the classical sign detector is applied. As with [2] and [3], the transformation we will employ will be summation, i.e. $Z_{1,n} = \sum_{i=1}^{n} Y_i$. Note that the "modified signal", if present, has strength ns. We will call such a detector a modified sign detector.

We will assume the noise process $\{X_i;i=1,2,\ldots\}$ is strong mixing with finite $2+\delta$ moment for some fixed $\delta > 0$, and with α-representation satisfying $\sum_{i=1}^{\infty} \alpha_i^{\delta/(2+\delta)} < \infty$. For simplicity we will assume the noise has continuous diagonally symmetric joint densities of all orders such that the asymptotic variance of $Z_{1,n}/\sqrt{n}$ is nonzero. Under such an assumption, the random variables $\{Z_{i,n};i=1,\ldots,N\}$ possess a symmetric univariate density under H_0 and admit application of the classical sign detector, resulting in the employment of the test statistic $T_s = \sum_{i=1}^{N} u(Z_{i,n})$, where $u(\cdot)$ is the indicator function of the nonnegative half line. Choosing n, m, and N for each N_0 is of course necessary when employing the above test statistic. This is addressed in the following section.

III. DEVELOPMENT

The nonparametric nature of the sign detector lies in its ability to fix a constant false alarm rate α over the class of admissible distributions. With a strong mixing noise process there will still be dependency present between the samples after the sample and skip procedure. Since fixing α is derived from an independence assumption, it is first necessary to show that the loss in the nonparametric nature of the modified sign detector can be controlled, i.e. the ratio $(\tilde{\alpha}-\alpha)/\alpha$ can be made arbitrarily small, where $\tilde{\alpha}$ is the actual false alarm rate. How this may be accomplished is answered in the following:

Theorem 1: Suppose the threshold and randomization probability are chosen so that the modified sign detector achieves false alarm probability α under an i.i.d. assumption on $\{Z_{i,n}; i=1,2,\ldots,N\}$. Then if $\tilde{\alpha}$ is the actual false alarm probability, we have $|(\tilde{\alpha}-\alpha)/\alpha| \leq 2\alpha^{N+1}\alpha_m$, where m samples are skipped between the $Z_{i,n}$.

Proof: Letting T_N be the threshold and r_N the randomization probability, we have

$$2^N\alpha = \sum_{j=T_N+1}^{N} \binom{N}{j} + \binom{N}{T_N} r_N, \quad \text{and}$$

$$\tilde{\alpha} = \sum_{j=T_N+1}^{N} P\{\{\#Z_{i,n} \geq 0\}=j\} + P\{\{\#Z_{i,n} \geq 0\}=T_N\}\, r_N .$$

An inductive argument may be used to show that

$$|P\{E_1 \cap \cdots \cap E_N\} - P\{E_1\}\cdots P\{E_N\}| \leq \alpha_m \sum_{i=0}^{N-2} 2^{-i},$$

where for each i, E_i is one of the two sets $\{\omega: Z_{i,n}(\omega) \geq 0\}$ or $\{\omega: Z_{i,n}(\omega) < 0\}$. Since there are $\binom{N}{j}$ possible choices of sequences of the E_i such that j of the E_i are of the form $E_i = \{\omega: Z_{i,n}(\omega) \geq 0\}$, and since $P(E_i) = 1/2$, we have

$$|\tilde{\alpha}-\alpha| \leq \alpha_m \sum_{i=0}^{N-2} 2^{-i} \left[\sum_{j=T_N+1}^{N} \binom{N}{j} + \binom{N}{T_N} r_N \right] \leq 2\alpha_m\, 2^N \alpha = 2^{N+1} \alpha_m \alpha .$$

Q.E.D.

For a preassigned tolerance ε_0, it is clear that if $\alpha_m \leq 2^{-(N+1)}\varepsilon_0$, then $|(\tilde{\alpha}-\alpha)/\alpha| \leq \varepsilon_0$. Because the major part of this paper will employ an asymptotic relative efficiency fidelity criterion (ARE), which has the advantage of allowing appeal to the Pitman-Noether theorem [5], we will assume in the sequel that $N \to \infty$ as $N_0 \to \infty$, since otherwise the test statistic is not asymptotically Gaussian as required in [5]. To properly define the ARE, we will allow the signal to approach zero via $s = K/\sqrt{N_0}$, where K is a positive constant. We will also limit consideration to those n,m,N for which we can apply [5], i.e. such that the limit $\lim_{N_0 \to \infty} Nn^2[f_n(0)]^2/N_0$ exists and is positive, where $f_n(\cdot)$ is the univariate density of $Z_{1,n}$. This assumption avoids a zero efficacy situation. We are now in a position to begin analysis of the asymptotic performance of the modified sign detector, where we assume tacitly that for each N_0, m is such that $\alpha_m \leq 2\alpha^{-(N+1)}\varepsilon_0$, which implies $(\tilde{\alpha}-\alpha)/\alpha \leq \varepsilon_0$. We first require the following central limit result, where $E_s\{\cdot\}$ denotes expectation when the signal is present with strength s.

Theorem 2: Suppose that under H_1 $s_{N_0} = K/\sqrt{N_0}$ and $s_{N_0} = 0$ under H_0, and let

$$\tilde{T}_{N_0} = (T_{s_{N_0}} - E_{s_{N_0}}\{T_{s_{N_0}}\})/(E_{s_{N_0}}\{(T_{s_{N_0}} - E_{s_{N_0}}\{T_{s_{N_0}}\})^2\})^{1/2}.$$

Then, under either hypothesis, as $N_0 \to \infty$ we have $\tilde{T}_{N_0} \xrightarrow{\mathscr{D}} N(0,1)$.

Proof: Note that if $n \not\to \infty$ as $N_0 \to \infty$, then by our regularity assumptions $\{X_i\}$ is m-dependent and the theorem is trivial. Assume $n \to \infty$ as $N_0 \to \infty$. First, consider the case where $\tilde{T}_{N_0}$ is under H_0, i.e. $s_{N_0} = 0$ for all N_0. Let $\varepsilon > 0$, and let $C_X(\cdot)$ denote the characteristic function for the random variable X. Note that in this case $\tilde{T}_{N_0} = \hat{T}_{N_0}\sqrt{N}/(4E_0\{(T_0 - N/2)^2\})^{1/2}$, where $\hat{T}_{N_0} = 2(T_0 - N/2)/\sqrt{N}$. Now $C_{\hat{T}_{N_0}}(t) = E_0\left\{\prod_{i=1}^{N} \exp(jtS(i,n,N))\right\}$, where $S(i,n,N) = 2(u(Z_{i,n}) - \frac{1}{2})/\sqrt{N}$. Letting $Y = \sum_{i=1}^{N-1} S(i,n,N)$, a straightforward modification of Lemma 1.3 of [6] to the complex case yields $|C_{\hat{T}_{N_0}}(t) - C_{S(N,n,N)}(t)\, C_Y(t)| \leq 40\,\alpha_m^{\delta/(2+\delta)}$, whence an iterative argument yields

$$|C_{\hat{T}_{N_0}}(t) - \prod_{i=1}^{N} C_{S(i,n,N)}(t)| \leq 40N\alpha_m^{\delta/(2+\delta)}$$

$$\leq 40N\left[2^{\delta/(2+\delta)}\right]^{-(N+1)} \varepsilon_0^{\delta/(2+\delta)},$$

and hence since $N \to \infty$ as $N_0 \to \infty$ there exists $\tilde{N}_0$ such that

$|C_{\hat{T}_{N_0}}(t) - \prod_{i=1}^{N} C_{S(i,n,N)}(t)| < \varepsilon/2$ for all $N_0 \geq \tilde{N}_0$. Now let $Z_1, Z_2, \ldots$ be a sequence of independent identically distributed $\{0,1\}$-valued random variables with $P(Z_i = 0) = 0.5$. Then $2(Z_1 + Z_2 + \ldots + Z_N - N/2)/\sqrt{N} \xrightarrow{\mathscr{D}} N(0,1)$, and $C_{2(Z_1+\ldots+Z_N-N/2)/\sqrt{N}}(t) = \prod_{i=1}^{N} C_{2(Z_i - \frac{1}{2})/\sqrt{N}}(t)$. Moreover, $C_{2(Z_i - \frac{1}{2})/\sqrt{N}}(t) = C_{S(i,n,N)}(t)$ for each i,n,N,t. Thus there exists $\hat{N}_0$ such that for all $N_0 \geq \hat{N}_0$ we have $|\prod_{i=1}^{N} C_{S(i,n,N)}(t) - C_X(t)| < \varepsilon/2$, where X is N(0,1). Therefore, for all $N_0 \geq \max\{\tilde{N}_0, \hat{N}_0\}$ we have $|C_{\hat{T}_{N_0}}(t) - C_X(t)| < \varepsilon$, which implies $\hat{T}_{N_0} \xrightarrow{\mathscr{D}} N(0,1)$. If we note that Lemma 1.3 of [6] implies $|1 - 4E_0\{(T_0 - N/2)^2\}/N| \leq 20(N-1)\,\alpha_m^{\delta/(2+\delta)} \leq 20(N-1)\left[2^{\delta/(2+\delta)}\right]^{-(N+1)} \varepsilon_0^{\delta/(2+\delta)}$, we obtain $4E_0\{(T_0-N/2)^2\}/N \to 1$. It then follows from [4, p. 25] that $\tilde{T}_{N_0} \xrightarrow{\mathscr{D}} N(0,1)$ under H_0. The general result follows by noting that Lemma 1.3 of [6] implies (where $Z_{i,n,0}$ denotes $Z_{i,n}$ under H_0)

$$E_{s_{N_0}}\left\{\left(\sum_{i=1}^{N}(u(Z_{i,n})-E_{s_{N_0}}\{u(Z_{i,n})\}) - \sum_{i=1}^{N}(u(Z_{i,n,0}) - \tfrac{1}{2})\right)^2/N\right\}$$

$$\le E_{s_{N_0}}\left\{(u(Z_{1,n})-E_{s_{N_0}}\{u(Z_{1,n})\}-u(Z_{1,n,0}) + \tfrac{1}{2})^2\right\} + 2(N-1)\ 10\alpha_m^{\delta/(2+\delta)}$$

$$\le \left[E_{s_{N_0}}\{(u(Z_{1,n})-u(Z_{1,n,0}))^2\}^{1/2} + \left|E_{s_{N_0}}\{u(Z_{1,n})\} - \tfrac{1}{2}\right|\right]^2 + 20(N-1)\left[2^{\delta/(2+\delta)}\right]^{-(N+1)} \varepsilon_0^{\delta/(2+\delta)}$$

$$= \left[\left(\int_{-ns_{N_0}}^{0} f_n(x)dx\right)^{1/2} + \int_{-ns_{N_0}}^{0} f_n(x)dx\right]^2 + 20(N-1)\left[2^{\delta/(2+\delta)}\right]^{-(N+1)} \varepsilon_0^{\delta/(2+\delta)}$$

Let $\tilde{F}_n(\cdot)$ denote the distribution function of $(X_1+\ldots+X_n)/\sqrt{n}$, and we see that $\int_{-ns_{N_0}}^{0} f_n(x)dx = \tilde{F}_n(0)-\tilde{F}_n(-\sqrt{n}s_{N_0}) = \tilde{F}_n(0)-\tilde{F}_n(-K\sqrt{n/N_0})$. Now $N = G((N_0+m)/(n+m)) \le N_0/n$ and $N\to\infty$ implies $\lim_{N_0\to\infty} n/N_0 = 0$. Let $\tilde{Y}_{N_0} = (X_1+\ldots+X_n)/\sqrt{n} + K\sqrt{n/N_0}$. Note that it follows from Theorem 1.7 of [6] that $(X_1+\ldots+X_n)/\sqrt{n} \xrightarrow{\mathscr{D}} N(0,\sigma^2)$ for some $\sigma > 0$. It thus follows from [4, p. 25] $\tilde{Y}_{N_0} \xrightarrow{\mathscr{D}} N(0,\sigma^2)$, which implies $\tilde{F}_n(0)-\tilde{F}_n(-K\sqrt{n/N_0}) \to 0$. We therefore obtain

$$E_{s_{N_0}}\left\{\left[2\sum_{i=1}^{N}(u(Z_{i,n})-E_{s_{N_0}}\{u(Z_{i,n})\})/\sqrt{N} - \hat{T}_{N_0}\right]^2\right\} \to 0,$$ and it follows from [4, p. 25] that $2\sum_{i=1}^{N}(u(Z_{i,n})-E_{s_{N_0}}\{u(Z_{i,n})\})/\sqrt{N} \xrightarrow{\mathscr{D}} N(0,1)$ under H_1. The final result then follows from [4, p. 25] since

$$\left|1-4E_{s_{N_0}}\left\{\left[\sum_{i=1}^{N}(u(Z_{i,n})-E_{s_{N_0}}\{u(Z_{i,n})\})\right]^2\right\}\Big/N\right|$$

$$\le \left|1-4[(1-\mu_n)^2\mu_n + \mu_n^2(1-\mu_n)]\right| + 20(N-1)\left[2^{\delta/(2+\delta)}\right]^{-(N+1)} \varepsilon_0^{\delta/(2+\delta)},$$

where $\mu_n = \int_{-ns_{N_0}}^{\infty} f_n(x)dx$. We see that this bound approaches zero since

$\int_{-n s_{N_0}}^{0} f_n(x)dx \to 0$ implies $\mu_n \to \frac{1}{2}$. Q.E.D.

We are now in a position to establish an upper bound on the performance of the modified sign detector. From the proof of Theorem 2 and our regularity assumptions it is routine to verify that conditions A-C of the hypothesis of the Pitman-Noether theorem [5] are satisfied, and condition D' follows directly from Theorem 2. Since [5] provides an expression for the ARE in terms of the efficacy functional $\eta(\cdot)$, it will suffice to provide a bound on $\eta(\cdot)$.

Theorem 3: Let $\eta(T_s)$ denote the efficacy of the modified sign detector. Then $\eta(T_s) \leq 4 \lim\limits_{N_0 \to \infty} \sup n[f_n(0)]^2$.

Proof: We have

$$\eta(T_s) = \lim_{N_0 \to \infty} \frac{\left[\frac{\partial}{\partial s} E_s \sum\limits_{i=1}^{N} u(Z_{i,n}) - N/2 \;\Big|_{s=0}\right]^2}{N_0 E_0\left\{\left(\sum\limits_{i=1}^{N} u(Z_{i,n}) - N/2\right)^2\right\}}$$

$$= \lim_{N_0 \to \infty} \frac{N^2 n^2 [f_n(0)]^2}{N_0 E_0\left\{\left(\sum\limits_{i=1}^{N} u(Z_{i,n}) - N/2\right)^2\right\}},$$

and thus we infer from Lemma 1.3 of [6] that (where $\eta = \eta(T_s)$)

$$\eta \leq \lim_{N_0 \to \infty} \sup \frac{N n^2 [f_n(0)]^2}{N_0\left(E_0\{(u(Z_{1,n}) - \frac{1}{2})^2\} - 20(N-1)\,\alpha_m^{\delta/(2+\delta)}\right)}$$

$= \lim\limits_{N_0 \to \infty} \sup 4Nn^2[f_n(0)]^2/N_0$ via evaluating α_m. But

$N = G((N_0+m)/(n+m)) \leq N_0/n$, and hence $\eta(T_s) \leq \lim\limits_{N_0 \to \infty} \sup 4n[f_n(0)]^2$. Q.E.D.

The proof of Theorem 3 also provides a representation for the efficacy which is expressed in the following corollary.

Corollary: We have $\eta(T_s) = \lim\limits_{N_0 \to \infty} 4Nn^2 f_n(0)^2/N_0$.

Proof: From the proof of Theorem 3 we obtain

$$\eta(T_s) \leq \lim_{N_0 \to \infty} \sup \frac{N n^2 [f_n(0)]^2}{N_0\left(E_0\{(u(Z_{1,n}) - \frac{1}{2})^2\} - 20(N-1)\,\alpha_m^{\delta/(2+\delta)}\right)}$$

and in a similar manner

$$\eta(T_s) \geq \liminf_{N_0 \to \infty} \frac{N n^2 [f_n(0)]^2}{N_0 \left(E_0\{(u(Z_{1,n})-\frac{1}{2})^2\} + 20(N-1)\, \alpha_m^{\delta/(2+\delta)}\right)},$$

from which the desired result follows via evaluating α_m. Q.E.D.

Note that in view of the corollary, the upper bound of Theorem 3 is achieved if $\lim_{N_0 \to \infty} Nn/N_0 = 1$. Also, recall from [3] that in the m-dependent case an ARE fidelity criterion generally pointed toward a choice of large block length n; we then note that the bound of Theorem 3 also bounds the efficacy of the detectors of [2-3] asymptotically in n. We would thus expect that the performance of an optimal detector for the strong mixing noise case would compare favorably to that designed under the much more restrictive m-dependence assumption. The following theorem shows that for a large class of noise processes there exist detectors with performance arbitrarily close to optimal.

Theorem 4: Suppose there exist constants a and r, $0 < r < 1$, such that $\alpha_i \leq a r^i$ for each i, and suppose that $\lim_{N_0 \to \infty} \sqrt{n}\, f_n(0)$ exists and is positive. Then for $\varepsilon_0, \varepsilon > 0$ there exists a modified sign detector T_s such that $|(\tilde{\alpha}-\alpha)/\alpha| \leq \varepsilon_0$ and $\eta(T_s) > 4 \lim_{N_0 \to \infty} n f_n^2(0) - \varepsilon$. In fact, T_s is obtained through $N = G(b\sqrt{N_0})$ and $m = c + dN$, for some constants b,c,d.

Proof: Note that since $\lim_{N_0 \to \infty} \sqrt{n}\, f_n(0)$ exists and is positive, the regularity assumption that $N n^2 [f_n(0)]^2 / N_0$ converges to a positive constant is satisfied if $\lim_{N_0 \to \infty} N n/N_0 > 0$. We have $|(\tilde{\alpha}-\alpha)/\alpha| \leq \varepsilon_0$ if $2^{N+1} \alpha_m \leq \varepsilon_0$, which is true if $m \geq [(\log \varepsilon_0 - \log a)/\log r - \log 2/\log r] - [\log 2/\log r]\, N$, and hence we take $c = G[(\log \varepsilon_0 - \log a)/\log r - \log 2/\log r] + 1$, $d = G[-\log 2/\log r] + 2$, where all logs are to the base e. Note that $d > 1$; thus $m \geq N$ for large N_0. We must show that the choice of m and N admits a choice of n for N_0 sufficiently large. This will occur if we can find integers n and x, with $0 \leq x < n+m$ and $n > 0$, such that $(N-1)(n+m) + n + x = N_0$, i.e. $n = (N_0 - (N-1)m)/N - x/N$. For $b < 1/\sqrt{d}$ and for N_0 sufficiently large we have $N_0 - (N-1)m \geq N$; moreover, if $(N_0 - (N-1)m)/N$ is not an integer there exists an integer x, $0 \leq x < N$ such that $(N_0 - (N-1)m)/N - x/N$ is an integer. But $N \leq m$ for large N_0 implies $0 \leq x < n+m$ and thus n exists if $b < 1/\sqrt{d}$ and N_0 is sufficiently large. In the sequel assume b is fixed such that $b < 1/\sqrt{d}$. We obtain, for sufficiently large N_0,

$$N \leq (N_0 + c + dN)/(n + c + dN) \leq N+1,$$

and thus

$$1 + \frac{c}{n} + \frac{dN}{n} \le \frac{N_0}{Nn} + \frac{c}{Nn} + \frac{d}{n} \le 1 + \frac{c}{n} + \frac{dN}{n} + \frac{1}{N} + \frac{c}{Nn} + \frac{d}{n}.$$

Now $n = (N_0 - (N-1)m)/N - x/N$ implies $n/N = [N_0 - (N-1)(c+dN)]/N^2 - x/N^2$, and thus

$$\lim_{N_0 \to \infty} \sup n/N = \lim_{N_0 \to \infty} \sup N_0/N^2 - d,$$

$$\lim_{N_0 \to \infty} \inf n/N = \lim_{N_0 \to \infty} \inf N_0/N^2 - d.$$

But $\lim_{N_0 \to \infty} N_0/N^2 = 1/b^2$, and hence $\lim_{N_0 \to \infty} n/N = 1/b^2 - d$. This implies $\lim_{N_0 \to \infty} N_0/(Nn) = 1 + db^2/(1-b^2d)$, which implies $\lim_{N_0 \to \infty} Nn/N_0 = 1 - b^2d$, and in view of the comments following the corollary to Theorem 3 the result is established by choosing $b < \min\{1/\sqrt{d}, \sqrt{\varepsilon/d} \big/ \big(2 \lim_{N_0 \to \infty} \sqrt{n}\, f_n(0)\big)\}$. Q.E.D.

Note that $\lim_{N_0 \to \infty} n/\sqrt{N_0} = 1/b - bd$, whereas $\lim_{N_0 \to \infty} N/\sqrt{N_0} = b$, and thus optimal performance is approached (as $b \to 0$) by choosing a relatively large block length and small number of blocks, although both are of order $\sqrt{N_0}$. This is therefore loosely consistent with the results of [3].

We remark finally that in the case where N remains bounded it might be best to employ a finite sample criterion analogous to that discussed in [3].

IV. CONCLUSION

We have shown how the modified sign detector may be designed to insure that the loss in its nonparametric nature is arbitrarily small. We have also established a bound on the asymptotic performance of the detector, and have shown that there exist detectors whose performance approaches this bound arbitrarily closely. We have also specified the form of such detectors.

ACKNOWLEDGEMENT

This research was supported by the Air Force Office of Scientific Research, Air Force Systems Command, USAF, under Grant AFOSR-76-3062, and by the Department of Defense Joint Services Electronics Program under Contract F49620-77-C-0101.

REFERENCES

1. J. B. Thomas, "Nonparametric detection," Proc. IEEE, vol. 58, pp. 623-631, May 1970.

2. S. A. Kassam and J. B. Thomas, "A class of nonparametric detectors for dependent input data," IEEE Trans. Inform. Theory, vol. IT-21, pp. 431-437, July 1975.

3. D. R. Halverson and G. L. Wise, "On the performance of a modified sign detector for m-dependent data," Proc. Seventeenth Annual Allerton Conf. on Communication, Control, and Computing, Monticello, Illinois, October 10-12, 1979.

4. P. Billingsley, Convergence of Probability Measures. New York: Wiley, 1968.

5. G. E. Noether, "On a theorem of Pitman," Ann. Math. Stat., vol. 26, pp. 64-68, 1955.

6. I. A. Ibragimov, "Some limit theorems for stationary processes," Theory Prob. Appl., vol. 7, pp. 349-382.

ROBUST DETECTION OF KNOWN SIGNALS IN ASYMMETRIC NOISE

SALEEM A. KASSAM, GEORGE MOUSTAKIDES AND JUNG GIL SHIN
Department of Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104

Department of Elect. Engineering
Stevens Inst. of Tech.
Hoboken, NJ 07030

ABSTRACT

The detection of signals in noise with possibly asymmetric probability density functions is considered. The noise density model allows a symmetric contaminated-nominal central part and an arbitrary tail behavior. For detection of known signals the robust likelihood-ratio (LR)-type detector is obtained, based on detector efficacy as performance criterion. The robust M-detector structure for constant-signal detection is also explicity obtained.

I. INTRODUCTION

Following the fundamental works of Huber on robust estimation [1] and robust hypothesis testing [2], many further developments and applications of robustness theory have been formulated by researchers in the communications sciences. Concepts of robustness in signal processing applications were certainly in existence prior to Huber's results [e.g., 3, 4]. However, it is generally accepted that the techniques and results in [1, 2] formed an important basis for much of the considerable subsequent research activity on robust schemes for signal estimation, detection and filtering applications. A recent survey paper [5] lists a large number of references on robust techniques.

In [6] Huber's ideas were applied to obtain the structures of asymptotically robust signal detectors. This resulted specifically in the canonical limiter-correlator detector for a weak deterministic signal in nominally Gaussian noise modeled as having a mixture or contaminated probability density. In [7] this result was extended to apply to other nominal noise densities. Both [6] and [7] considered detection structures of the type where the sum of memoryless transformations of each discrete-time input observation (the test statistic) is compared to a fixed threshold. For example, the limiter-correlator robust detector for a signal vector $(s_1, s_2, \ldots s_n)$ in an observation vector $(X_1, X_2, \ldots X_n)$ with independent and identically distributed additive noise computes the test statistic $T_n = \sum_{i=1}^{n} L_i(X_i)$, where $L_i(X_i) = s_i \ell(X_i)$ and ℓ is a limiter characteristic. We will call such detector structures LR-detectors, since this is the usual structure for a likelihood-ratio test on independent observations. More recently Huber's results were used in [8, 9] to obtain directly the robust M-detectors for both the fixed-sample and sequential cases. An M-detector structure is obtained when the detection test statistic Q_n is obtained as that function of the observations minimizing $\sum_{i=1}^{n} M(X_i - s_i Q_n)$, where M is some appropriately chosen function. Note that Q_n may be used as an estimator for the signal amplitude, and such an estimator is called an M-estimator because of its similarity to maximum likelihood estimators in general.

Two major factors limit the applicability of such results for signal processing schemes. One of these is the requirement of independence for the sequence of discrete-time input data to the detectors. This requirement of independence has recently been addressed in [10], where it was shown that robust detector structures can be derived for operation under conditions of

---

This research is supported by contracts NRL N00014-78-C-0798 and ONR N00014-80-K-0945.

weak dependence in the input sequence. The results in [10] were developed for detection applications following similar considerations which had earlier been applied in [11, 12] for robust estimation. The second main limitation of many previous results on robust detection has been the assumption that the allowable noise density functions are symmetric. In this paper we will be concerned with this latter problem, and we will develop the structures of the robust LR- and M-detectors for robust detection of weak deterministic signals with a noise model allowing asymmetry in the univariate noise density functions. Our study was largely motivated by some recent work on robust estimation with asymmetrically distributed noise [13, 14]; in particular, we will adapt and draw upon the techniques and results in [14] for this work.

Specifically, we consider the class $\hat{\mathcal{F}}_{g,\varepsilon,d}$ of noise densities f given by

$$\hat{\mathcal{F}}_{g,\varepsilon,d} = \{ f \mid f = \begin{cases} (1-\varepsilon)g + \varepsilon h, \text{ on } [-d,d] \\ \text{arbitrary, outside } [-d,d] \end{cases}, h \in \hat{\mathcal{H}} \} \tag{1}$$

Here $\varepsilon \in (0,1)$ is the maximum degree of contamination of a nominal density function g. The density g is assumed to be strongly unimodel (i.e., -log g is convex) and symmetric, and in addition we assume it to be sufficiently regular so that g' is absolutely continuous. The parameter d is a positive parameter specifying the interval around the origin in which the noise density f is a bounded, symmetric, contaminated version of g. The class $\mathcal{H}$ is the class of all bounded, symmetric, density functions. Note that a valid $f \in \mathcal{F}_{g,\varepsilon,d}$ could be zero on $(-\infty,-d)$, and place a probability of $2(1-\varepsilon)[1-G(d)] + \varepsilon$ on (d,∞), where G is the distribution function corresponding to g. If g is the zero-mean Gaussian density with variance σ^2, a reasonable specification of d may be a number between 2σ and 4σ, and ε is typically between 0.001 and 0.1.

In the next sections we will consider the robust LR- and M-detectors for the noise model of (1).

II. ROBUST LR-DETECTOR FOR ASYMMETRIC NOISE DENSITIES

For a vector of observations $(X_i, X_2, \ldots, X_n)$ of length n, described by

$$X_i = N_i + \theta s_i \quad , \qquad i = 1,2,\ldots,n$$

where $(s_i, s_2, \ldots, s_n)$ is a deterministic signal vector and the N_i are independent and identically distributed noise components, we want to test the null hypothesis H_0 that $\theta=0$ versus the alternative H_1 that $\theta > 0$. For an LR-detector using test statistic

$$T_n = \sum_{i=1}^{n} s_i \ell(X_i),$$

we want to obtain the characteristic ℓ which results in a robust detector, for allowable univariate noise density functions f in the class $\mathcal{F}_{g,\varepsilon,d}$. As a criterion of performance we will use the detector efficacy $\mathcal{E}(f,\ell)$ which is dependent on f and ℓ, and defined as [15]

$$\mathcal{E}(f,\ell) = \lim_{n\to\infty} \frac{[\frac{d}{d\theta}E_\theta\{T_n\}|_{\theta=0}]^2}{\mathrm{var}_\theta\{T_n\}|_{\theta=0}} \tag{2}$$

It is clear that for an LR-detector to be consistent for all $f \in \hat{\mathcal{F}}_{g,\varepsilon,d}$, the characteristic ℓ has to vanish outside $[-d,d]$. Since $\hat{\mathcal{F}}_{g,\varepsilon,d}$ consists of densities symmetric on $[-d,d]$, we additionally require that the allowable ℓ are symmetric. Let $\mathcal{L}_c$ denote the class of all LR-detector characteristics ℓ satisfying

i) $\ell(x) = 0, \; |x| \geq c$,

ii) $\ell(x) = -\ell(-x)$,

iii) ℓ is absolutely continuous,

the parameter c being a non-zero cut-off value, $c \leq d$; the value of c is set by consistency requirements, as we will discuss soon.

Solution for Efficacy-Robust Detector

We want to find a least-favorable density f_R in $\hat{F}_{g,\varepsilon,d}$ and a corresponding optimum characteristic ℓ_R in L_c such that

$$\inf_{f \in \hat{F}_{g,\varepsilon,d}} E(f,\ell_R) = E(f_R,\ell_R) \tag{3}$$

Note that we will then have

$$E(f_R,\ell_R) = \sup_{\ell \in L_c} E(f_R,\ell) \tag{4}$$

since we require ℓ_R to be optimum for f_R.

The following theorem establishes the condition under which a pair (f_R,ℓ_R) can be found in $\hat{F}_{g,\varepsilon d} \times L_c$ satisfying (3) and (4) with a finite, non-zero value $E(f_R,\ell_R)$. It is directly related to Theorem 3.1 in [14].

Theorem 1: If the condition

$$\varepsilon < (1-\varepsilon)\{2cg(0) - [2G(c)-1]\} \tag{5}$$

is satisfied, where G is the distribution function corresponding to g, the density function

$$f_R(x) = \begin{cases} (1-\varepsilon)g(x) & , |x| \leq a_0 \\ \dfrac{(1-\varepsilon)g(a_0)}{\cosh^2[\frac{1}{2}a_1(c-a_o)]} \cosh^2[\frac{1}{2}a_1(c-|x|), & a_0 < |x| \leq c \\ \text{arbitrary} & , |x| > c \end{cases} \tag{6}$$

and the corresponding optimum characteristic in L_c

$$\ell_R(x) = \begin{cases} \dfrac{-f'_R(x)}{f_R(x)} & , |x| \leq c \\ 0 & , |x| > c \end{cases} \tag{7}$$

satisfy (3) and (4), with $0 < a_0 < c$ and $\dfrac{-g'(a_0)}{g(a_0)} < a_1$ being the unique solutions of

$$\varepsilon = \int_{-c}^{c} f_R(x)dx - (1-\varepsilon)\int_{-c}^{c} g(x)dx$$

and

$$\frac{-g'(a_0)}{a_1 g(a_0)} = \tanh[\tfrac{1}{2}a_1(c-a_0)]$$

Comments on Theorem 1. If condition (5) is not satisfied, then f_R can be picked to be a constant on [-c,c] so that $E(f_R,\ell)=0$. In [14, Table 1] and in Table 1, some numerical values are given for the upper bounds $\varepsilon(c)$ on ε, for different c, for which (5) is satisfied with g the unit variance Gaussian density. The value of $\varepsilon(c)$ is larger than 0.1 if c is larger than unity. The proof of Theorem 1 is a direct extension of the proof of Theorem 3.1 in [14], which placed more restrictive conditions on the allowable density functions and estimator characteristics. Our criterion of detection efficacy is used as an estimator criterion in [14]; however, its interpretation as an estimation variance for the statistics in [14] requires further assumptions which are not simply characterized as conditions on the allowable density functions. We will elaborate on this in the next section.

For $c \leq d$, it can be easily shown that ℓ_R of (7) results in a test which is consistent for θ in a positive neighbourhood of the origin. To see this, it is sufficient to show that the slope at $\theta=0$ of the mean function of ℓ_R, given by

$$m'(0) = \int_{-c}^{c} \ell_R'(x) f(x) dx,$$

is positive for all $f \in \hat{F}_{g,\varepsilon,d}$. Now under the condition of Theorem 1, $E(f_R, \ell_R)$ is positive, and $\int_{-c}^{c} \ell_R'(x) f_R(x) dx$ is positive. Since $\hat{F}_{g,\varepsilon,d}$ is convex, it follows that $m'(0) > 0$ for all $f \in \hat{F}_{g,\varepsilon,d}$.

Now even if $m'(0)$ is positive, the mean function $m(\theta) = \int_{-c}^{c} \ell_R(x) f(x-\theta) dx$ will become zero for some positive value θ_{max} of θ. In Table II we show the computed value of θ_{max} as a function of c and ε, for the case c=d, with g again the unit-variance Gaussian density. These values were obtained by considering for each θ the noise density minimizing $m(\theta)$. These results indicate that for given ε, increasing d with c=d leads to a test consistent for an increasing range of values of θ; in the limit, we get consistency for all θ and the solution degenerates to that in [6,7].

It has not been possible to prove a stronger result which simultaneously bounds the worst-case asymptotic local slope of the power function and the false-alarm probability as in [6,7], where the fixed threshold of the robust LR-detector could also be determined. The reason why the stronger result is not possible here is that ℓ_R is not a monotone limiter characteristic, so that the numerator and denominator in (2) are not separately minimized and maximized, respectively, by f_R when $\ell = \ell_R$. In the next section we consider an M-detector robust structure which allows a stronger robustness property to be derived.

III. ROBUST M-DETECTOR FOR ASYMMETRIC NOISE DENSITIES.

We will restrict attention to the special case of constant signals, and without further loss of generality we will take $s_i=1$, $i=1,2,\ldots n$. The general case of non-constant signals requires further considerations, as we will indicate later. Our M-detectors will therefore be based on statistics Q_n satisfying

$$\sum_{i=1}^{n} \psi(X_i - Q_n) = 0, \tag{8}$$

so that we are implicitly assuming M to be sufficiently regular.

We are interested in the class $F_{g,\varepsilon,d}$ (or a useful subset of the class $\hat{F}_{g,\varepsilon,d}$) of densities, and we impose the following reasonable constraints in defining the class Ψ_c of allowable M-detector functions ψ we will consider:

(i) $\psi(x) = 0, \ |x| \geq c,$
(ii) $\psi(x) = -\psi(-x),$
(iii) $\psi'(x)$ is bounded and piecewise-continuous on $[-c,c]$.

An additional constraint will be added soon. The value of the parameter $c < d$, which defines the size of Ψ_c, is set by requirements we consider next.

To complete the specification of our class of M-detectors a solution scheme has to be specified for obtaining Q_n satisfying (8). This is necessary because a soluiton to (8) may not be unique. The scheme we specify is a simpler iterative procedure than that considered in [14]. Although its numerical convergence rate to the solution may be somewhat smaller, it allows us to obtain more explicit robustness results than were obtained in [14] (specifically, note remark 3.5 in [14]).

We define the test-statistic Q_n of an M-detector based on $\psi \in \Psi_c$ in terms of the sequence $\{Q_n^j\}$ given by

$$Q_n^{j+1} = Q_n^j + \frac{\lambda_n(Q_n^j)}{D}, \quad j=0,1,2,\ldots, \tag{9}$$

where

$$\lambda_n(q) = \int_{-\infty}^{\infty} \psi(x-q)dF_n(x), \tag{10}$$

F_n being the empirical distribution function of the n observations, with D being a positive constant. The solution Q_n is defined as $Q_n = \lim_{j\to\infty} Q_n^j$, with Q_n^0 a sample median (or any consistent estimator of the median of the X_i), provided the limit exists. Otherwise Q_n is taken to be Q_n^0.

In terms of the quantities g, ε and d defining $\hat{F}_{g,\varepsilon,d}$ we now define a new parameter k_0 by

$$k_0 = G^{-1}\left[\frac{1}{2} + \frac{\tau}{2(1-\varepsilon)}\right] \tag{11}$$

with

$$\tau = 2(1-\varepsilon)\ [1-G(d)] + \varepsilon, \tag{12}$$

or directly as

$$k_0 = G^{-1}\left[1-G(d) + \frac{1}{2(1-\varepsilon)}\right] \tag{13}$$

G being the distribution function corresponding to the density g. Note that τ is the maximum value of the total probability which may be distributed, arbitrarily, outside [-d,d]. We will assume that our model is such as to make $\tau < 0.5$, so that the median of any $f \in \hat{F}_{g,\varepsilon,d}$ lies in (-d,d). This implies the restriction $\varepsilon < 0.5$. From (12) we have

$$d = G^{-1}\left[\frac{1}{2} + \frac{1-\tau}{2(1-\varepsilon)}\right],$$

so that $k_0 < d$. Under this assumption ($\tau<0.5$) the maximum value of $|m|$, where m is the median of $f \in \hat{F}_{g,\varepsilon,d}$, satisfies

$$[(1-\varepsilon)[G(|m|)-G(-d)] = \frac{1}{2}$$

so that

$$G(|m|) = 1-G(d) + \frac{1}{2(1-\varepsilon)},$$

and thus the median is always in $[-k_0,k_0]$, from (13).

Our objective now is to establish conditions for the asymptotic normality of Q_n based on $\psi \in \Psi$ for $f \in \hat{F}_{g,\varepsilon,d}$. We will finally be able to define subsets of Ψ_c and $\hat{F}^c_{g,\varepsilon,d}$ over which asymptotic normality holds, and we will obtain the saddlepoint solution for the asymptotic variance over these classes. The following results are stated without proofs; proofs will be given in a detailed version of the paper [16].

We consider first the consistency of Q_n.

Lemma 1. Let $c \leq d - k_o$ and ψ be any characteristic in Ψ_c. Suppose that for a given $f \in \hat{F}_{g,\varepsilon,d}$ the median lies in the open interval $(-k_0,k_0)$ and the function

$$\lambda(q) = \int_{-c+q}^{c+q} \psi(x-q)f(x)dx$$

is strictly decreasing on $[-k_0,k_0]$. Then for $D > \frac{1}{2}\max_{[-c,c]}|\psi'(x)|$ in (9), the M-detector based on ψ has for this f a test statistic Q_n which converges in probability to θ; in addition,

$$\lim_{n\to\infty} P\{\sum_{i=1}^{n} \psi(X_i-Q_n) = 0\} = 1$$

Note that this lemma implies that the iterations for Q_n in (9) will converge (and therefore Q_n is not defined as Q_n^0) with a probability approaching unity as $n \to \infty$.

The following lemma is concerned with asymptotic normality of Q_n.

Lemma 2. Let (ψ,f) be a pair in $\Psi_c \times \hat{F}_{g,\varepsilon,d}$, with $c \leq d - k_0$, for which the conditions of Lemma 1 are satisfied. If in addition we have $\lambda'(q) < 0$ in a closed neighbourhood of the origin, then $\sqrt{n}(Q_n-\theta)$ is asymptotically normally distributed with variance $V(f,\psi)$ given by

$$V(f,\psi) = \frac{1}{E(f,\psi)} = \frac{\int_{-c}^{c}\psi^2(x)f(x)dx}{[\int_{-c}^{c}\psi'(x)f(x)dx]^2}$$

Lemma 2 follows from results in [17, Section 4] where general conditions are given ensuring asymptotic normality of M-estimators. That these conditions are met under Lemma 2 can be easily demonstrated.

We are now ready to obtain a least-favorable density and corresponding optimum M-detector characteristic which together form a saddlepoint for performance in terms of asymptotic variance of a consistent and asymtotically normal detection test statistic. First we define $\psi_R \in \Psi_c$ by $\psi_R(x) = \ell_R(x)$ of (7), where $f_R(x)$ was defined in (6). Then we have

Lemma 3. Let g, ε, d be such that $\tau < \frac{1}{2}$ in (26) and k_0 of (13) is less than $\frac{a_0}{2}$, for $c \leq d - k_0$ satisfying (5). [The parameter a_0 is present in the definition of $\psi_R = \ell_R$ in (7).] Then for $f = f_R$ and $\psi=\psi_R$ the test statistic Q_n obtained from (9) with $D > \frac{1}{2}\max_{[-c,c]}|\psi_R'(x)|$ is consistent for θ, and $\sqrt{n}(Q_n-\theta)$ is asymptotically normally distributed with variance

$$V(f_R,\psi_R) = \frac{\int_{c}^{-c}\psi_R^2(x)f_R(x)dx}{[\int_{-c}^{c}\psi_R(x)f_R'(x)dx]^2} \qquad (14)$$

Let $\mathcal{F}^*_{g,\varepsilon,d} \subset \hat{\mathcal{F}}_{g,\varepsilon,d}$ be the subset of densities in $\hat{\mathcal{F}}_{g,\varepsilon,d}$ which are strictly unimodal on $[-d,d]$. In addition, the parameters ε and d are restricted to satisfy the condition $\tau < 0.5$, with τ defined in (12), and the medians are assumed to lie in $(-k_0,k_0)$, with k_0 defined in (13). This last condition requires positive probabilities to be distributed both on $(-\infty,-d)$ and on (d,∞), and is an insignificant restriction. Finally, with $c = d-k_0$, the parameter a_0 defining f_R in (6) is assumed to be larger than $2k_0$ and also (5) is assumed to be satisfied. It is easy to show that these conditions are satisfied for reasonable choices of g, ε, and d. For example, let g be the zero-mean, unit-variance Gaussian density, let $\varepsilon = 0.05$ and $d = 2$. Then $\tau = 0.093$ and $k_0 = 0.123$. With $c = d-k_0 = 1.877$ (5) is satisfied, and the value of $\frac{a_0}{2}$ is 0.502.

We finally restrict consideration to the subset $\Psi^*_c \subset \Psi_c$, containing M-detector characteristics which are non-negative on $(0,c)$ in addition to satisfying the three conditions defining ψ_c. This is a reasonable restriction, in view of the strict unimodality restriction used in defining $\mathcal{F}^*_{g,\varepsilon,d}$.

With the specification of g, ε, and d implicitly defining $c = d-k_0$, we get the following result:

Theorem 2. (i) Let $\psi_R \in \Psi^*_c$ be defined by $\psi_R = \ell_R$ of (7), and let Q_n be the test statistic arising from the M-detector based on ψ_R, with $D > \frac{1}{2}\max_{[-c,c]} |\psi_R'(x)|$. Then for any $f \in \mathcal{F}^*_{g,\varepsilon,d}$, Q_n is a consistent estimator for θ; $\sqrt{n}(Q_n-\theta)$ is asymptotically normally distributed with variance $V(f,\psi_R)$ satisfying

$$\max_{f \in \mathcal{F}^*_{g,\varepsilon,d}} V(f,\psi_R) = V(f_R,\psi_R),$$

where $V(f_R,\psi_R)$ was defined in (14).

(ii) For any $\psi \in \Psi^*_c$ which gives a consistent statistic for all $f \in \mathcal{F}^*_{g,\varepsilon,d}$,

$$V(f_R,\psi_R) \leq V(f_R,\psi)$$

where $V(f_R,\psi)$ is the asymptotic variance of the normalized statistic based on ψ.

Theorem 2 provides the main result of this section. We started from classes $\hat{\mathcal{F}}_{g,\varepsilon,d}$ and Ψ_c of densities and detector characteristics, respectively, but our least-favorable density and robust characteristic were obtained from amongst functions in $\mathcal{F}^*_{d,\varepsilon,g}$ and Ψ^*_c, respectively. The essential restriction added was that the densities considered be strictly unimodal on $[-d,d]$; this is not unreasonable, because the nominal density g is assumed to have this property. In applying the robust statistic Q_n arising from ψ_R in signal detection, one has to set a threshold based on the maximum variance $V(f_R,\psi_R)$. Thus the false-alarm probability constraint is automatically satisfied. In addition, for any $f \in \mathcal{F}^*_{g,\varepsilon,d}$, the asymptotic power function or the slope of the power function at the origin can be lower bounded by the corresponding values for $f = f_R$, depending on specific conditions on the signal strength parameter and detection threshold values. The details can be found in [8]; the main condition has been proved in Theorem 2 (specifically, the condition in [8, Lemma 2]).

The major reason why we confined attention to the constant-signal case in this section is that we need a reasonable initial value (e.g., the median Q_n^0) in starting the iterations in (9), to guarantee a consistent statistic Q_n. Extension of our results to the general case seems possible, and would appear to require an initial estimate of the median based on some nonparametric or other simple regression procedure [18,19]. The consistency proofs and conditions would also have to be extended.

IV. CONCLUSION

We have derived the structures of robust LR- and M-detectors for known-signal detection in noise for which the probability density has a symmetric, contaminated central part and arbitrary tail behavior. This model has been used previously for robust estimation studies.

The robust LR-detector was derived for performance defined by detection efficacy, a weak-signal large sample-size asymptotic performance measure. Although the detection efficacy is directly related to the slope of the detector power function, it was not possible to obtain simultaneous control on the false-alarm probability.

For constant-signal detection, the robust M-detector characteristic was obtained for performance characterized by the asymptotic variance of the test statistic. This result for the asymptotic variance, together with earlier results on M-detectors, allow more interesting robust detection solutions which can maintain the false-alarm probability within desired upper bounds. Our robustness results differ from previous results on robust estimation which also examined asymptotic variance in that we considered a simpler solution strategy and obtained the explicit saddlepoint solution for well-defined classes of noise and detector characteristic.

REFERENCES

[1] P.J. Huber, "Robust estimation of a location parameter," Ann. Math Stat., vol. 35, pp. 73-101, 1964.

[2] P.J. Huber, "A robust version of the probability ratio test," Ann. Math. Stat., vol. 36, pp. 1753-1758. 1965.

[3] M.C. Yovits and J.L. Jackson, "Linear filter optimization with game theory considerations," IRE National Conv. Rec., pt. 4, pp. 193-199, 1955.

[4] W.L. Root, "Communications through unspecified additive noise," Information and Control, vol. 4, pp. 15-29, March 1961.

[5] A.A. Ershov, "Stable methods of estimating parameters," Automation and Remote Control, vol. 39, No. 7, pt. 2, pp. 1152-1181, July 1978.

[6] R.D. Martin and S.C. Schwartz, "Robust detection of a known signal in nearly Gaussian noise," IEEE Trans. Information Theory, vol. IT-17, pp. 50-56, Jan. 1971.

[7] S.A. Kassam and J.B. Thomas, "Asymptotically robust detection of a known signal in contaminated non-Gaussian noise," IEEE Trans. Information Theory, vol. IT-22, pp. 22-26, Jan. 1976.

[8] A.H. El-Sawy and V.D. VandeLinde, "Robust detection of known signals," IEEE Trans. Information Theory, vol. IT-23, pp. 722-727, Nov. 1977.

[9] A.H. El-Sawy and V.D. VandeLinde, "Robust sequential detection of signals in noise," IEEE Trans. Information Theory, vol. IT-25, pp. 346-353, May 1979.

[10] H.V. Poor, "On detection in weakly dependent noise," Proc. 1979 Conf. on Inform. Sciences and Systems, (Johns Hopkins), pp. 499-504, March 1979.

[11] S.L. Portnoy, "Robust estimation in dependent situations," Ann. of Stat., vol. 5, pp. 22-43, 1977.

[12] S.L. Portnoy, "Further remarks on robust estimation in dependent situations," Ann. of Stat., vol. 7, pp. 224-231, 1979.

[13] L.A. Jaeckel, "Robust estimates of location: symmetry and asymmetric contamination," Ann. Math. Stat., vol. 42, pp. 1020-1034, 1971.

[14] J.R. Collins, "Robust estimation of a location parameter in the presence of asymmetry," Ann. of Stat., vol.14, pp. 68-85, 1976.

[15] J. Capon, "On the asymptotic efficiency of locally optimum detectors," IRE Trans. Information Theory, vol. IT-7, pp. 67-71, April 1961.

[16] S.A. Kassam, G. Moustakides and J.G. Shin, "Robust detection of known signals in asymmetric noise," submitted for publication.

[17] P.J. Huber, "The behavior of maximum likelihood estimates under non-standard conditions," Proc. Fifth Berkeley Symp. on Math. Stat. and Prob., vol. 1, pp. 221-233, 1967.

[18] R.V. Hogg, "Statistical robustness: one view of its use in applications today, "The Amer. Statistician, vol. 33, pp. 108-115, August 1979.

[19] D.F. Andrews, "A robust method for multiple linear regression," Technometrics, vol. 16, pp. 523-531, Nov. 1974.

c	ε (c)
1.0	0.103
1.5	0.248
2.0	0.391
2.5	0.502
3.0	0.583

TABLE I. Upper Bound on ε as Function of c for which (5') holds, for Unit-Variance Gaussian Density g.

ε \ c=d	1.0	1.5	2.0	2.5	3.0
0.05	0.0	1.7	2.7	3.5	4.0
0.10	0.0	0.9	2.4	3.0	3.5
0.15	---	0.0	2.0	2.7	3.2
0.20	---	0.0	1.5	2.3	2.8
0.25	---	---	1.2	1.9	2.4
0.30	---	---	0.9	1.6	2.1
0.35	---	---	0.5	1.3	1.8
0.40	---	---	---	0.8	1.5

TABLE II. Upper Bound θ_{max} on θ as Function of c, ε (with c=d) for which Robust, LR-Detector is Consistent. (Unit-Variance Gaussian g.)

Performance Analysis of an Optimum Detector with Counting Point Process Observations

Aurel A. Lazar and Stuart C. Schwartz[†]

Columbia University and Princeton University

ABSTRACT

Transformations of the classical hypothesis testing problem with continuous-time observations to a hypothesis testing problem with counting point process observations are considered. A general form for the random intensity rate (RIR) which can accommodate feedback is investigated and the optimal solution in the Neyman-Pearson sense is specified. An analysis of the performance of the corresponding optimum processor is studied.

I. Introduction

In [1] and [2] we considered the classical hypothesis testing problem

$$\begin{aligned} &H_0\text{: under } \bar{P}_0,\ (X_t),\ 0\leqslant t\leqslant T, \quad \text{is a Wiener process,} \\ &H_1\text{: under } \bar{P}_1,\ (X_t-\int_0^t S_u\,du),\ 0\leqslant t\leqslant T, \quad \text{is a Wiener process,} \end{aligned} \tag{1}$$

where $X=(X_t,F_t\otimes\tilde{F}_0)$, $0\leqslant t\leqslant T$, are the continuous-time observations, and investigated transformations of this test into the hypothesis testing problem

$$\begin{aligned} &H_0\text{: under } \bar{P}_0,\ (N_t),\ 0\leqslant t\leqslant T, \text{ is a DSPP with RIR } (\hat{\lambda}_t^{(0)}),\ 0\leqslant t\leqslant T, \\ &H_1\text{: under } \bar{P}_1,\ (N_t),\ 0\leqslant t\leqslant T, \text{ is a DSPP with RIR } (\hat{\lambda}_t^{(1)}),\ 0\leqslant t\leqslant T, \end{aligned} \tag{2}$$

where $N=(N_t,F_0\otimes\tilde{F}_t)$, $0\leqslant t\leqslant T$, are the counting point process observations. In addition, several suboptimum detection schemes were presented.

The aim of the investigations in [1] and [2] was to find, from among a given class of codes, the RIR λ as a functional of the "information bearing process" X and the observations N, *i.e.*,

$$\lambda_t = \lambda_t(X,N), \tag{3}$$

such that we attain or come close to some "optimum properties".

In this paper we extend the results of [1] and [2] by finding, from the class of o-admissible codes, the Neyman-Pearson optimum RIR λ. The paper is organized as follows. In section II we consider the classical "input" hypothesis testing problem of deciding between two absolutely continuous measures $\bar{P}_0$ and $\bar{P}_1$ ($\bar{P}_1 << \bar{P}_0$), given some observation process $X=(X_t,F_t\otimes\tilde{F}_0)$, $0\leqslant t\leqslant T$. In the sequel the corresponding Radon-Nikodym derivative, denoted by Λ_T, will be called the input likelihood ratio. We also consider the "output" hypothesis testing problem of deciding between the probability measures $\bar{P}_0$ and $\bar{P}_1$ given the counting point process observations $N=(N_t,F_0\otimes\tilde{F}_t)$, $0\leqslant t\leqslant T$. The corresponding Radon-Nikodym derivative will be called the output likelihood ratio. Theorem 1 specifies the Neyman-Pearson optimum RIR (of significance level α_0) for detection in the class of o-admissible codes. It is shown that, if the causality condition on λ is dropped, the optimal RIR is given by

---

[†] Presently on leave at the Radio Research Laboratory, Bell Telephone Laboratories, Holmdel, New Jersey.

$$\lambda_t=\mu_0+b\cdot\chi_{[0,L]}(\Lambda_T), \tag{4}$$

for all t, $0\leqslant t\leqslant T$, where b is constant and $\chi_{[0,L]}$ is the characteristic function of the set $[0,L]$. We conjecture that if λ is nonanticipative the RIR is given by (for more details see section II)

$$\lambda_t=\mu_0+b\cdot\chi_{[0,L_t]}(\Lambda_t), \tag{5}$$

for all t, $0\leqslant t\leqslant T$.

The degradation which appears in the output detection problem by using the optimum RIR λ (introduced by the counting point process observations) is established in section III. The main result in section III is presented in Theorem 2. It relates (see also Lemma 3) the power and the probability of false alarm of a very general hypothesis testing problem when its sample space is not directly observable (DSPP observations) to the same parameters α and β one would obtain if the input space were directly observable.

II. The Optimal Random Intensity Rate for Detection

Let us consider the "input" hypothesis testing problem of deciding between two absolutely continuous probability measures $\bar{P}_0$ and $\bar{P}_1$ ($\bar{P}_1<<\bar{P}_0$), defined on the measurable space $(\bar{\Omega},\bar{F})$, given the observation process $X=(X_t,F_t\otimes\bar{F}_0)$, $0\leqslant t\leqslant T$. The construction of the probability measures $\bar{P}_0$ and $\bar{P}_1$ is given in [3]. The corresponding Neyman-Pearson test of significance level α_0 is completely characterized by the sufficient statistic $\chi_{[0,L]}(\Lambda_T)$, where L is the associated threshold [4].

Let us consider the "output" hypothesis testing problem of deciding between the probability measures $\bar{P}_0$ and $\bar{P}_1$ given the observation counting point process $N=(N_t,F_0\otimes\tilde{F}_t)$, $0\leqslant t\leqslant T$. The corresponding Neyman-Person test of significance level α_0 is completely characterized by $\tilde{\chi}_{[0,\tilde{L}]}(\tilde{\Lambda}_T)$ where $\tilde{L}$ is the threshold .

Definition 1. The class of codes satisfying the peak and average power constraint

$$0\leqslant\lambda_t(X,N)\leqslant c \qquad \text{and} \qquad \bar{E}_i\frac{1}{T}\int_0^T\lambda_s ds\leqslant qc, \tag{6}$$

for all t and i, $0\leqslant t\leqslant T$ and $i=0,1$ and constants c and q, $c\in R_+$ and $0<q\leqslant 1$ is called *o-admissible.*

Definition 2. The coding $\lambda=(\lambda_t(X,N),F_t\otimes\tilde{F}_t)$, $0\leqslant t\leqslant T$, is said to be *optimum in the Neyman-Pearson sense of significance level* α_0 if over the class of o-admissible codes the supremum

$$\sup_{0\leqslant\lambda_t\leqslant c,\,\bar{E}_i\frac{1}{T}\int_0^T\lambda_s ds\leqslant qc}\tilde{\beta} \tag{7}$$

is achieved for all α, $\alpha\leqslant\alpha_0$, i.e., the coding λ maximizes the power among all Neyman-Pearson tests.

Remark. The optimal RIR λ in the Neyman-Pearson sense features the best performance for only a given probability of false alarm α_0. Therefore the above optimality criterion leads in general to solutions which are "locally" optimal. In other words for each α_0 a different code λ might be obtained.

Due to analytical difficulties we will restrict the following analysis to the case where λ_t is measurable with respect to the σ-algebra $F_T \otimes \tilde{F}_t$ for all t, $0 \leqslant t \leqslant T$.

Theorem 1. *The RIR* $\lambda = (\lambda_t, F_T \otimes \tilde{F}_t)$, $0 \leqslant t \leqslant T$, *corresponding to the (output observation) counting point process* $\overline{N}$ *is optimum in the Neyman-Pearson sense if it is given by*

$$\lambda_t(X,N) = \mu_0 + b \cdot \chi_{[0,L]}(\Lambda_T), \tag{8}$$

where $b = (1 \wedge \frac{q}{\beta \vee \alpha_0})c$, *for all* t, $0 \leqslant t \leqslant T$.

A proof of this theorem appears in [3].

Remark. The above theorem implies that, under the Neyman-Pearson optimality criterion, feedback does not increase the power of the output hypothesis testing problem. This result is somewhat similar to the result obtained on the capacity of the Poisson type channel under a peak and average power constraint (see also [5]-[7]).

Since

$$\hat{\lambda}_s^{(0)} = \frac{1}{1 + \frac{\alpha}{1-\alpha} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_{s-}}} \tag{9}$$

and

$$\hat{\lambda}_s^{(1)} = \frac{1}{1 + \frac{\beta}{1-\beta} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_{s-}}}, \tag{10}$$

for all s, $0 \leqslant s \leqslant T$, we conclude that the test statistic for the output hypothesis testing problem is given by

$$\tilde{\Lambda}_T \left\{ \begin{array}{lll} \geqslant \tilde{L} & \rightarrow & H_1 \\ < \tilde{L} & \rightarrow & H_0, \end{array} \right. \tag{11}$$

where

$$\tilde{\Lambda}_T = \int_0^T \ln \frac{\mu_0 + \frac{b}{1 + \frac{\beta}{1-\beta} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_{s-}}}}{\mu_0 + \frac{b}{1 + \frac{\alpha}{1-\alpha} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_{s-}}}} dN_s$$

$$- b \int_0^{\mathrm{T}} [\frac{1}{1 + \frac{\beta}{1-\beta} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_s}} - \frac{1}{1 + \frac{\alpha}{1-\alpha} e^{bs} (\frac{\mu_0}{\mu_0 + b})^{N_s}}] ds. \tag{12}$$

For the case where λ_t is measurable with respect to the σ-algebra $F_t \otimes \tilde{F}_t$, for all t, $0 \leqslant t \leqslant T$, we state the following

Conjecture 1. *The RIR* $\lambda=(\lambda_t, F_t \otimes \tilde{F}_t)$, $0 \leqslant t \leqslant T$, *corresponding to the (output observation) counting point process* N *is optimum in the Neyman-Pearson sense if it is given by*

$$\lambda_t(X,N) = \mu_0 + b \cdot \chi_{[0,L_t]}(\Lambda_t), \tag{13}$$

where $b=(1 \wedge \frac{q}{\beta \vee \alpha_0})c$, *for all* t, $0 \leqslant t \leqslant T$. *The threshold* $L: R_+ \rightarrow R_+$ *is defined by*[†]

$$\bar{P}_0(\Lambda_t \geqslant L_t) = \alpha_0.$$

Remark. The RIR λ described by (13) can be seen as the causal version of the RIR λ given by (8).

In the next section an analysis of the performance of the optimum detection scheme is presented. It is shown that an equivalent form of (12) can be given that allows a simple analysis of the performance of the above optimum detection scheme.

III. Performance Analysis of the Optimal Detection Scheme

The performance analysis of the optimum Neyman-Pearson detector (12) is greatly simplified by the following:

Lemma 1. *If* $\beta \geqslant \alpha$ and $\lambda_s = \mu_0 + b \cdot \chi(\Lambda_T)$ *for every* s, $0 \leqslant s \leqslant T$, *then*[†]

$$\tilde{\Lambda}_T = \tilde{\Lambda}_T(N_T), \tag{14}$$

where $\tilde{\Lambda}_T$ *is a nonincreasing Borel function of* N_T.

A proof of this lemma based on the abstract Bayes formula [8] is given in [3].

Since $\tilde{\Lambda}_T = \tilde{\Lambda}_T(N_T)$ is a nonincreasing function, the Neyman-Pearson test

$$\tilde{\Lambda}_T \begin{cases} \geqslant \tilde{L} & \rightarrow \quad H_1 \\ < \tilde{L} & \rightarrow \quad H_0, \end{cases} \tag{15}$$

(where $\tilde{L}$ is the threshold of the output test) is equivalent to

$$N_T \begin{cases} < l & \rightarrow \quad H_1 \\ \geqslant l & \rightarrow \quad H_0 \end{cases} \tag{16}$$

(where l is the corresponding new threshold of the output test).

Since $\tilde{\Lambda}_T$ is a nonincreasing function of the observations, the power $\tilde{\beta}$ and the level $\tilde{\alpha}$ of the output hypothesis testing problem with counting point process observations can be reduced to

$$\begin{aligned} \tilde{\beta} &= \bar{P}_1(N_T < l) = \sum_{k=0}^{l-1} \bar{P}_1(N_T = k), \\ \tilde{\alpha} &= \bar{P}_0(N_T < l) = \sum_{k=0}^{l-1} \bar{P}_0(N_T = k), \end{aligned} \tag{17}$$

[†] Randomized tests are not considered here. The extension to randomized tests can be done in the usual manner [4].

where l is the new threshold.

In the following, with straightforward mathematics, we will find an equivalent expressions for $\tilde{\beta}$ and $\tilde{\alpha}$. Our standing assumption is that the observation process $N=(N_t, F_0 \otimes \tilde{F}_t)$, $0 \leqslant t \leqslant T$, is a doubly stochastic Poisson process (DSPP).

Let us first relate the power β and the probability of false alarm α of the input to the power $\tilde{\beta}$ and probability of false alarm $\tilde{\alpha}$ of the output. We can give the following

Lemma 2.

$$\begin{aligned}
\tilde{\beta} &= \sum_{k=0}^{l-1} \frac{[(\mu_0+b)]^k}{k!} \exp[-(\mu_0+b)T] - \sum_{k=0}^{l-1} \frac{T^k}{k!} [(\mu_0+b)^k \exp(-bT) - \mu_0^k] \exp(-\mu_0 T)\beta, \\
\tilde{\alpha} &= \sum_{k=0}^{l-1} \frac{[(\mu_0+b)]^k}{k!} \exp[-(\mu_0+b)T] - \sum_{k=0}^{l-1} \frac{T^k}{k!} [(\mu_0+b)^k \exp(-bT) - \mu_0^k] \exp(-\mu_0 T)\alpha.
\end{aligned} \tag{18}$$

Proof. Both $\tilde{\beta}$ and $\tilde{\alpha}$ can be obtained with some easy manipulations, and only those for $\tilde{\beta}$ are presented.

$$\tilde{\beta} = \sum_{k=0}^{l-1} \bar{P}_1(N_T=k) = \sum_{k=0}^{l-1} \frac{T^k}{k!} \bar{E}_1 \{ (\mu_0 + b \cdot \chi_{[0,L]}(Z))^k \exp[-(\mu_0 + b \cdot \chi_{[0,L]}(Z))T] \}$$

$$= \sum_{k=0}^{l-1} \frac{T^k}{k!} \{ \mu_0^k \exp(-\mu_0 T) + [(\mu_0 + b)^k \exp[-(\mu_0+b)T] - \mu_0^k \exp(-\mu_0 T)](1-\beta) \}$$

$$= \sum_{k=0}^{l-1} \frac{[(\mu_0+b)]^k}{k!} \exp[-(\mu_0+b)T] - \sum_{k=0}^{l-1} \frac{T^k}{k!} [(\mu_0+b)^k \exp(-bT) - \mu_0^k] \exp(-\mu_0 T)\beta. \qquad \square$$

Let us compare the pairs (β, α) and $(\tilde{\beta}, \tilde{\alpha})$ in the limiting case $\mu_0=0$. The test statistic can be easily reduced to

$$N_T \begin{cases} = 0 \rightarrow & H_1 \\ \geqslant 1 \rightarrow & H_0. \end{cases} \tag{19}$$

The power $\tilde{\beta}$ and the probability of false alarm $\tilde{\alpha}$ can also be computed directly:

$$\begin{aligned}
\tilde{\beta} &= \bar{P}_1(N_T=0) = e^{-bT} + (1 - e^{-bT})\beta, \\
\tilde{\alpha} &= \bar{P}_0(N_T=0) = e^{-bT} + (1 - e^{-bT})\alpha.
\end{aligned} \tag{20}$$

Fig. 1 illustrates graphically the results that we might expect in terms of the receiver operating characteristic (ROC). Since $\beta = \tilde{\beta}$ implies $\beta=1$, the only point of intersection of the above curves occurs for $(\alpha, \beta) = (1,1)$.

Another useful representation for $\tilde{\alpha}$ and $\tilde{\beta}$ is given in

Lemma 3.

$$\begin{aligned}
\beta &= 1 - \bar{E}_1 \chi_{[0,L]}(Z), \\
\tilde{\beta} &= 1 - \bar{E}_1 \{ \sum_{k \geqslant l} \frac{(\mu_0 T)^k}{k!} \exp(-\mu_0 T) - \sum_{k=0}^{l-1} \frac{T^k}{k!} [(\mu_0+b)^k \exp(-bT) - \mu_0^k] \exp(-\mu_0 T) \cdot \chi_{[0,L]}(Z) \},
\end{aligned} \tag{21}$$

[†] By abuse of notation.

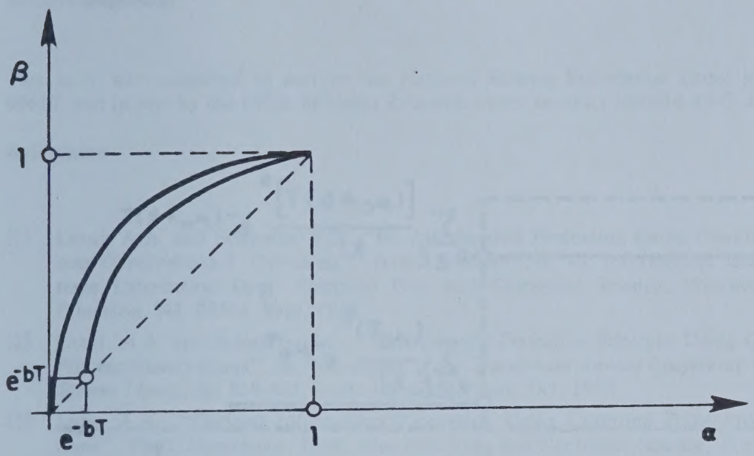


Fig. 1 The receiver operating characteristic (optimum processor).

$$\begin{aligned}\alpha &= 1 - \bar{E}_0\, \chi_{[0,L]}(Z), \\ \tilde{\alpha} &= 1 - \bar{E}_0\{\sum_{k \geqslant l} \frac{(\mu_0 T)^k}{k!} \exp(-\mu_0 T) - \sum_{k=0}^{l-1} \frac{T^k}{k!} [(\mu_0 + b)^k \exp(-bT) - \mu_0^k] \exp(-\mu_0 T) \cdot \chi_{[0,L]}(Z)\},\end{aligned} \tag{22}$$

where $\chi_{[0,L]}(Z)$ *denotes the characteristic function of the set* $[0,L]$ *on the real line.*

Proof. The proof can be easily supplied. □

If we define the function $\kappa_{opt}: R_+ \rightarrow [0,1]$ by

$$\kappa_{opt}(\zeta) = \sum_{k \geqslant l} \frac{T^k}{k!} \mu_0^k \exp(-\mu_0 T)$$

$$- \sum_{k=0}^{l-1} \frac{T^k}{k!} \{(\mu_0 + b)^k \exp[-(\mu_0 + b) T] - \mu_0^k \exp(-\mu_0 T)\} \cdot \chi_{[0,L]}(\zeta), \tag{23}$$

for all ζ, $\zeta \in R_+$, we arrive at the following fundamental result:

Theorem 2. *The optimum detector at the output of the processor*[†] *performs a randomized test given by*

$$\kappa_{opt}(\zeta), \tag{24}$$

which is an approximation to the optimum input test specified by

$$\chi_{[0,L]}(\zeta), \tag{25}$$

for all ζ, $\zeta \in R_+$, respectively. The limiting case where $\mu_0 = 0$ can easily be analyzed. We have

$$\begin{aligned}\tilde{\beta} &= 1 - \bar{E}_1 \chi_{[0,L]}(Z)(1 - e^{-bT}), \\ \tilde{\alpha} &= 1 - \bar{E}_0 \chi_{[0,L]}(Z)(1 - e^{-bT}),\end{aligned} \tag{26}$$

[†] See [1]-[3] for the definition of the processor.

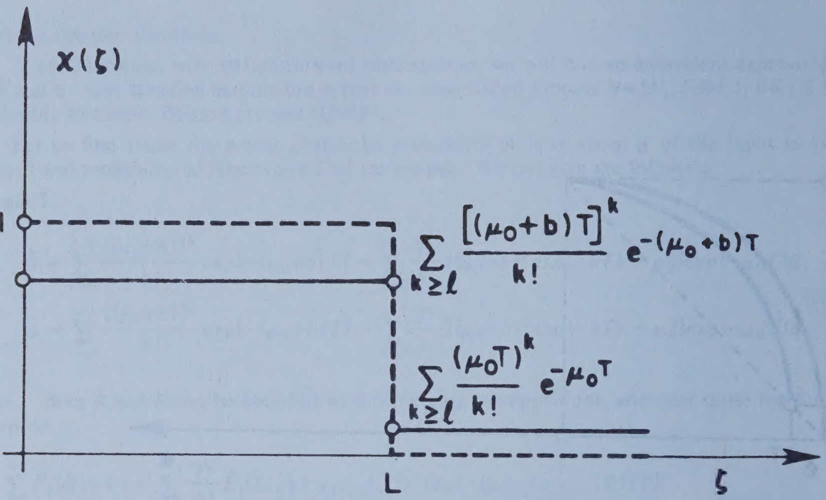


Fig. 2 Representation of the optimum input and optimum output test.

and thus the test statistic is given by

$$\kappa_{opt}(\zeta) = (1 - e^{-bT})\chi_{[0,L]}(\zeta), \tag{27}$$

for all ζ, $\zeta \in R_+$. The results obtained are depicted in Fig.2.

The decision procedure at the output of the processor is strictly suboptimal when compared to the optimal decision procedure at the input, should the input be completely observable. This is true since, under the Neyman-Pearson criterion, any randomized test will not perform as well as the corresponding optimum test given by (25) [9, p.65]. This agrees with our intuitive feeling that the transformation which takes place through the processor (see also Fig.2.1 in [3]) affects the information.

Remark. The randomized test κ_{opt} has a very simple probabilistic interpretation. According to this interpretation, for a particular value of the input likelihood ratio ζ, with probability $\kappa_{opt}(\zeta)$ we decide that the null-hypothesis is true and with probability $1 - \kappa_{opt}(\zeta)$ we decide that the alternative is true. The deviation between the optimum randomized test $\chi_{[0,L]}$ and the randomized test κ_{opt} will define a measure for evaluating the performance of the processor.

IV. Conclusion

In this paper the Neyman-Pearson optimum RIR λ from among the class of o-admissible codes has been specified. Since the problem of finding the ROC even for simple hypothesis testing problems is known to be analytically intractable, a new type of comparison between the optimum and a suboptimum processor has been introduced. In particular, the output Neyman-Pearson test compared to the similar test for the input turns out to be a randomized test. The latter test shows us to what degree the processor specified by the RIR λ leads to a loss of "information". This loss supports our intuitive feeling that the "stochastic mapping" describing the processor has to reduce the amount of "information" available at its input.

Acknowledgement

This work was supported in part by the National Science Foundation under grant ENG-75-09610, and in part by the Office of Naval Research under contract N00014-80-C-0530.

References

[1] Lazar, A.A. and Schwartz, S.C., "On Information Processing Using Counting Point Process Observations I: Detection", *Technical Report No. 44*, Information Sciences and Systems Laboratory, Dept. Electrical Eng. and Computer Science, Princeton University, Princeton, NJ 08544, Sept. 1979.

[2] Lazar, A.A. and Schwartz, S.C., "Suboptimum Detection Schemes Using Counting Point Process Observations", in *Proceedings of the Seventeenth Annual Conference on Circuits and System Theory*, pp. 858-865, Univ. Illinois, Urbana, Oct. 1979.

[3] Lazar A.A., "Optimal Information Processing Using Counting Point Process Observations", *Ph.D. Dissertation*, Dept. Electrical Eng. and Computer Science, Princeton University, Princeton, NJ 08544, Oct. 1980.

[4] Silvey, S.D., *Statistical Inference*, Chapman and Hall, London, 1975.

[5] Kabanov, YU.M., "The Capacity of a Channel of the Poisson Type", *Theory of Probability and its Applications*, Vol. **23**, pp. 143-147, Jan. 1978.

[6] Davis, M.H.A., "Capacity and Cutoff Rates for Poisson Type Channels", preprint, 1979.

[7] Lazar, A.A., "On the Capacity of the Poisson Type Channel", in *Proceedings of the Fourteenth Conference on Information Sciences and Systems*, Princeton University, Princeton, NJ 08544, March 1980.

[8] Liptser, R.S. and Shiryayev, A.N., *Statistics of Random Processes I: General Theory*, Springer Verlag, New York, 1977.

[9] Lehmann, E., *Testing Statistical Hypothesis*, John Wiley, New York, 1959.

A NEW CONCEPT OF MULTIPLE-PARAMETER QUANTUM MECHANICAL ESTIMATION WITH APPLICATIONS IN OPTICAL COMMUNICATIONS

Drago Dragoslav Ilic
Department of Electrical Engineering
and Computer Sciences
University of California at San Diego
and
General Director of CALTEX Investments
and Developments Corporations

Abstract

An entirely new concept of minimum error variance quantum estimation is developed. The approach is based on spectral calculus formulated on Banach algebra of operators over an appropriate Hilbert space. A new theorem, fundamental in theory of simultaneous quantum observations in optical communications, is presented and applied.

1. INTRODUCTION

It has been established that the channel capacity of the communication system can be significantly increased by increasing the carrier frequency. This introduced a great interest into optical communication systems. The developments in quantum electronics and super-high-frequency technology, as well as the invention of lasers and masers, have made possible information transmission at optical frequencies. As one could expect, new, inherently "non-classical" engineering problems arise. At optical frequencies, the quantum energy $\hbar\omega$ becomes comparable to the thermal energy kT, and the quantum nature of electromagnetic signals becomes essential. The classical detection, estimation, and filtering theory cannot be applied for construction of an optimal receiver or estimator at optical frequencies because in designing systems and devices for quantum signal processing one must go beyond such classical limitations as thermal noise or communication channel fading. The designer of an optimal quantum receiver has to consider additional "non-classical" limitations imposed by the quantum-statistical concept of measurement. Therefore, in order to develop the quantum theory of optimal signal processing, one has to generalize the methods of classical statistics, optimal control, stochastic filtering, and stochastic control. Currently, these generalizations are in the process of development and refinement. Additional motivation for quantum estimation theory can be found in (1,3,6,7,9,10,11).

A solution of the quantum estimation problem enables one to evaluate the potentialities of optical communication systems and to formulate the

the inherent limitations on their performance imposed by the quantum-statistical nature of the received electromagnetic field. Unfortunately, until recently there did not exist a general procedure for solving this problem. As pointed out in (12), in order to solve a quantum estimation problem it was necessary to guess a solution on the basis of symmetry, intuition, or experience, and then if possible to verify the optimality. Even if one approximates the density operators involved by finite square matrices and discretizes the parameter space X, there did not exist a method for solving the problem. According to the writer's knowledge, the first general method for solving the multi-parameter quantum problem appears in (13). The problem of estimation of the complex amplitude of a coherent signal in a harmonic oscillator, representing a single-mode of a quantum receiver with thermal noise, is the only multi-parameter problem of a practical importance in optical communications whose solution is known. The solution to that problem is exactly a solution of the procedure developed in (13). The method in (13) is entirely different from the one presented in this paper. Yet another approach to the quantum estimation problem is proposed in (3). It remains to be investigated which of the procedures is computationally the most attractive.

2. PROBLEM FORMULATION

Let $(\tilde{Y},G,q)$ be a probability space. Consider an N-dimensional random variable $x: (\tilde{Y},G) \rightarrow (X,F)$, and let (X,F,p) be the "range probability space" induced by the measurable vector-valued function $x(y)$, $y \in \tilde{Y}$. As usual, $L_2(X,F,p)$ will denote the Hilbert space of real-valued functions $g: X \rightarrow R$ that are locally square integrable with respect to the measure p. Let $x = (x_1,x_2,\ldots,x_N)$. Since we are interested in a minimum variance estimator,

$$x_i \in L_2(X,F,P), \qquad \forall\, i \in J_N := \{1,\ldots,N\}.$$

Consider a quantum-mechanical system Q (e.g. electron, photon, etc.) described by the density operator $Z_s(x)$ influenced in some measurable way by a random vector x. In optical communications Q represents the electromagnetic field in the aperture of a quantum receiver. A probability operator measure (p.o.m.) is defined as in (1,6,7,9,10,11).

Denote by $(X,F(X))$ the measurable parameter space, and by $(U,F(U))$ the measurable space of decisions, $U \subseteq R^n$. Let $S(H_s;F(U))$ be the set of all p.o. measures (quantum observables) $W:F(U) \rightarrow B_s(H_s)$. Introduce the functional $J:S(H_s;F(U)) \rightarrow R$,

$$J(W) := \mathrm{Tr}\int_X \int_U Z_s(x)\; Q(x,u)\; dW(u)\; dp(x),$$

where H_s is a separable complex Hilbert space which quantum mechanical formalism associates with each quantum system, and $B_s(H_s)$ denotes the self-adjoint elements of the Banach algebra of operators in H_s. Let

$$Q(x,u) := \sum_{i=1}^{n} (x_i - u_i)^2$$

represent the error cost. This type of error cost is the most frequently used in optical communications. The integral $J(W)$ has been rigorously defined as a Bochner integral in (8,9).

Minimum Variance Quantum Estimation Problem (MVQEP):

Find $W^\circ \in S(H_s;F(U))$ such that

$$J(W^\circ) \leq J(W), \qquad \forall W \in S(H_s;F(U)).$$

In the sequel we will utilize Holevo's and Davies's interpretation of Naimark's theorem [2,5; 6, pp. 78], which asserts that, given any quantum observable $W \in S(H_s;F(U))$, there exists an ancillary quantum system in a pure state Z_A, independent of x, and a projection-valued quantum observable $E:F(U) \rightarrow B_s(H_s \otimes H_A)$ such that

$$\mathrm{Tr}[Z_s(x)W(S)] = \mathrm{Tr}_{S+A}[(Z_s(x) \otimes Z_A)E(S)], \qquad \forall S \in F(U),$$

for any state $Z(x)$. The mapping $Z_s(x) \mapsto \mathrm{Tr}_{S+A}[(Z_s(x) \otimes Z_A)\; dW(u)]$, $u \in U$, is injective and provides the full description of the statistics of the measurement. The physical interpretation of a quantum measurement represented by a p.o.m. is given in (6,10,11). The ordered pair $(E;Z_A)$ corresponding to W, will be called a realization pair of W.

Define the functional

$$J(E;Z_A) := \int_X \int_U Q(x,u)\mathrm{Tr}_{S+A}[Z_s(x) \otimes Z_A)dE(u)]dp(x).$$

As demonstrated in (1)

$$J(E;Z_A) = J(W), \qquad \forall W \in S(H;F(U)).$$

Denote by $Y(W)$ the set of all realization pairs of W, and let Y be the set-theoretic union of $Y(W)$ over all $W \in S(H_s;F(U))$, i.e. $Y = \bigcup_W Y(W)$.

Problem 1. Find a pair $(\tilde{E};\tilde{Z}_A)$ such that

$$\inf_{Y} J(E;Z_A) = J(\tilde{E};\tilde{Z}_A).$$

Remark: Due to space limitation it is not possible to present the proofs. All statements appearing in this paper are rigorously demonstrated in (1).

The following lemma establishes the equivalence between Problem 1 and MVQEP.

Lemma 1. A solution to Problem 1 exists. A necessary and sufficient condition for W° to be a solution to MVQEP is that the corresponding realization pair $(E^\circ;Z_A^\circ)$ is a solution to Problem 1.

Let $(\tilde{E};\tilde{Z}_A)$ be a solution to Problem 1.

Denote

$$Z(x) := Z_s(x)\otimes\tilde{Z}_A; \qquad H := H_s\otimes H_A.$$

Introduce the self-adjoint operators

$$A^s := \underset{\sim}{E}_x[Z_s(x)],\ B_i^s := \underset{\sim}{E}_x[x_i Z(x)],\ A := A^s\otimes Z_A,\ B_i := B_i^s\otimes Z_A,\ i\in J_N,$$

where $\underset{\sim}{E}_x$ denotes expectation with respect to x.

Let K be a spectral measure in B(H) defined on (S,F(S)), where $S\subset R$ denotes the support of K, and F(S) the system of Borel subsets of S.

Introduce the linear manifold of real-valued functions

$$M(K) := \{f:R\to R \mid f \text{ is a Borel function, K-a.e. defined, K-measurable, finite K-a.e., K-integrable}, \int f^2(t)\mathrm{Tr}[AdK(t)] < \infty\}.$$

Consider the following problem:

Let $T\in B_s(H)$. Assume that T is capable of representing a dynamic variable of quantum system Q, and let K be the canonical spectral measure of T. Define the functional

$$J^i(f_i) := \int_X\int_S (x_i - f_i(t))^2\mathrm{Tr}[Z(x)dK(t)]dp(x). \tag{1}$$

$J^i(f_i)$ is well defined everywhere on M(K).

Problem 2: Find $\hat{f}_i : R\to R$, such that

$$\inf_{f_i \in M(K)} J^i(f_i) = J^i(\hat{f}_i). \tag{2}$$

In the foregoing problem we are not concerned how good an estimator T is. Our concern is the map f_i, and we ask the question: if T is any observable of Q, and if the quantum measurement represented by T yields the outcome t, what is the best postprocessing of the obtained data in the sense of relation (2) if the objective is optimal estimation of x_i. The density operator Z(x) is influenced by a vector-valued random variable x. However, we want to estimate only one component of x, say x_i, and we measure only one operator T. Everything is conventional and there is no mention of "simultaneous quantum-measurement".

The following lemma has been proved in two independent ways in (1):

Fundamental Lemma (FL). A solution to Problem 2 exists in M(K), is unique up to an equivalence class, and is characterized by

$$\hat{f}_i(t) = \frac{\mathrm{Tr}[B_i dK(t)]}{\mathrm{Tr}[A dK(t)]}, \qquad \forall t \in S.$$

Remark: The functional (1) is in the minimum variance form, and one would expect $\hat{f}_i(t) = \underset{\sim}{E}_x[x_i \mid t]$. Indeed, as proved in (1),

$$\underset{\sim}{E}_x[x_i \mid t] = \frac{\mathrm{Tr}[B_i dK(t)]}{\mathrm{Tr}[A dK(t)]} = \hat{f}_i(t).$$

The following theorem is a simplified version of one of von Neumann's theorems (4):

Let $\left\{F_i\right\}_{i=1}^{n}$ be a collection of permuting operators in H. Then there exists $T \in B_s(H)$ such that

$$F_i = f_i(T), \qquad \forall i \in J_n,$$

for some $f_i \in M(K)$, where K is the canonical spectral measure of T.

Define the functional

$$G(T;Z_A) = \underset{\sim}{E}_x\left\{\sum_{i=1}^{n} \mathrm{Tr}_{S+A}[Z(x)\ (x_i \underset{\sim}{1} - \hat{f}_i^T\ (T))^2]\right\}, \qquad T \in B_s(H)$$

where

$$\hat{f}_i{}^T(t) = \underset{\sim}{E}_x[x_i \mid t] = \frac{\mathrm{Tr}_{S+A}[B_i dK(t)]}{\mathrm{Tr}_{S+A}[A\, dK(t)]}, \qquad \forall t \in S\ .$$

Let

$$C := \{Q \varepsilon B_s(H_A) \mid Q^2 = Q, \quad tr_A Q = 1\} .$$

Problem 3. Find a pair $(\hat{T}, \hat{Z}_A) \varepsilon B_s(H_A) x C$ such that

$$\inf_{C} \inf_{B_s(H)} G(T;Q) = G(\hat{T};\hat{Z}_A) .$$

Lemma 2. Solution to Problem 3 exists.

Theorem 1. Problem 3 is equivalent to MVQEP.

Proof: The theorem follows from the foregoing von Neumann's theorem and the Fundamental Lemma of this paper.

Remark: The equivalence in the foregoing theorem is in the following sense: Let K be the canonical spectral measure of $\hat{T}$ and $F(R^N)$ the Borel σ-algebra of subsets of R^N. Note that Lemma 1 asserts that the functions $\hat{f}_i^T$ are Borel functions. Let $g_i := (\hat{f}_i^{\hat{T}})^{-1}$. Define the spectral measure $E°: F(R^N) \rightarrow P(H)$, where P(H) is a non-Boolean lattice of projection operators in H, by

$$E°(S) = \prod_{i=1}^{N} K \circ g_i(S_i) \qquad \forall S = \prod x S_i, \; S_i \varepsilon F(R) .$$

Introduce the p.o.m.

$$W°(S) := Tr_A[(\underset{\sim}{1}_s \otimes Z_A) \; E°(S)] , \qquad S \varepsilon F(R^N) .$$

Then, $W°(\cdot)$ is a solution to MVQEP.

Theorem 1, together with the Fundamental Lemma, is the major result of this paper. It says that the MVQEP can be reduced to Problem 3. Problem 3 consist of two optimization steps: (i) optimization over $B_s(H)$, and (ii) optimization over C. Note that the optimal quantum estimators are: $\hat{f}_i^{\hat{T}}(\hat{T})$, $i \varepsilon J_N$. Therefore, these estimators necessarily permute and all constrains inherent to simultaneous observations in quantum mechanics are imbedded in Problem 3. Thus the optimization step (i) is an unconstrained optimization problem over a C*-algebra and, evidently, belongs to the class of quite convenient optimization problems. Step (ii), which involves optimization over C, is trivial. Consequently, the extremely difficult quantum estimation problem is reduced to Problem 3 which can be solved by elementary optimization techniques. For example, since both the domain and the range space of the functional $f(\cdot;Q)$ are normed, we can use the Fréchet-derivative approach that is the simplest minimization technique that one could conceive for a solution of MVQEP.

Due to space limitation the further analysis of Problem 3 cannot be included in this paper. The detailed perturbation analysis of Problem 3 is given in (1). A discretized finite dimensional version of MVQEP, together with all appropriate examples, is presented in (15).

References

1. Ilic, D.D., An operational calculus approach to multiparameter quantum estimation problem. To appear.

2. Davies, E.B., Quantum Theory of Open Systems, Academic Press, New York (1976).

3. Ilic, D.D., A measure theoretic approach to simultaneous quantum estimation problem. To appear.

4. von Neumann, J. Uber Funktionen von Funktionaloperatoren, Ann. of Math. 32, (1931).

5. Ilic, D.D. and Tarn, T.J., Nonlinear quantum stochastic filter for discrete time systems. Proc. 4th Inter. Symp. on the Mathematical Theory of Networks and Systems, Delft, The Netherlands, July 3-6, 1979. Invited paper.

6. Helstrom, C.W., Quantum Detection and Estimation Theory, Academic Press, New York (1976).

7. Ilic, D.D., Optimization over σ-algebras -- A new concept of a nonlinear stochastic filter with application in optical communications with quantum measurements. Proc. 22nd Symp. on Circuits and Systems, Philadelphia, June 17-19, 1979.

8. Holevo, A.S., Statistical decision theory for quantum systems. J. Multivar. Anal. 3, 335-394 (Dec. 1973).

9. Ilic, D.D., Simultaneous Quantum Mechanical Observations and Nonlinear Quantum Stochastic Filter. Doctoral thesis, Dept. for Systems Science and Math., School of Engineering, Washington U.

10. Helstrom, C.W., "Simultaneous measurement" from the standpoint of quantum detection and estimation theory. Found. Phys. 4, 453-463 (Dec. 1974).

11. Helstrom, C.W. and Kennedy, R.S., Noncommuting observables in quantum detection and estimation theory. Trans. IEEE IT-20, 16-24 (Jan. 1974).

12. Helstrom, C.W., An introduction to quantum estimation theory. Multivariate Analysis -- IV, P.A. Krishnaiah, ed., North-Holland, Amsterdam (1977), 371-385.

13. Ilic, D.D., Minimum norm approach to multiparameter quantum estimation problem. Proc. 23rd Symp. on Circuits and Systems. Toledo, Ohio, Aug. 4-5, 1980. Invited paper.

14. Ilic, D.D., Simultaneous quantum mechanical measurements in optical communications with applications in stochastic filtering. To appear.

15. Ilic, D.D., A new estimation theory in optical communications with quantum receivers. Proc. Inter. Conference on Communications, Denver, Colorado, June, 1981.

VITA

- Received the B.S. and M.S. degrees in Electrical Engineering and Computer Sciences from the University of Belgrade.

- Received the M.S. and D.Sc. degrees in System and Computer Sciences from the School of Engineering of Washington University, St. Louis.

- During the period 1971-1975 worked, as a self-employed computer consultant, in Italy and England on special purpose computer hardware and multiprocessing.

- During the period 1976-1978 worked as Acting Director for Research and Development with the Computer Composition Sales Corporation from St. Louis.

- Since September 1978, with Department of Electrical Engineering and Computer Sciences, University of California at San Diego.

- General Director of TEXCAL Investments and Development Corporations from San Diego, California and Houston, Texas since June, 1980.

A NEW ROBUST ESTIMATOR OF LOCATION PARAMETER

Carlos A. Pomalaza and Clare D. McGillem
School of Electrical Engineering, Purdue University
West Lafayette, IN 47907

SUMMARY

Let $x_1, \ldots x_n$ be an independent identically distributed random sample with distribution $F(x-\theta)$. It is desired to estimate the location parameter θ. The estimator proposed here is of the form,

$$\hat{\theta} = \frac{\sum_{i=1}^{n} \psi(n_i) x_i}{\sum_{i=1}^{n} \psi(n_i)} \tag{1}$$

where $\psi(\cdot)$ is a positive increasing function and n_i = number of x's in $[x_i - c, x_i + c]$, c is a constant.

Define

$$F_n(t) = [\text{number of x's} \leq t]/n \,, -\infty < t < \infty$$
$$= \text{empirical c.d.f. of the sample}$$

If, for example, $\psi(n_i) = n_i$, (1) becomes

$$\hat{\theta} = \frac{\int t[F_n(t+c) - F_n(t-c)]dF_n(t)}{\int [F_n(t+c) - F_n(t-c)]dF_n(t)} \tag{2}$$

as $n \to \infty, \hat{\theta}$ tends to the functional

$$T(F) = \frac{\int t[F(t+c) - F(t-c)]dF(t)}{\int [F(t+c) - F(t-c)]dF(t)} \tag{3}$$

The influence curve $IC_{T,F}(x)$ can be now computed from (3). Specific values for the constant c of the estimator can be obtained by using the fact that, usually, $\sqrt{n}[T(F_n) - T(F)]$ is asymptotically normal with mean 0 and variance $\int [IC_{T,F}(x)]^2 dF(x)$. Numerical results obtained by Monte-Carlo simulations give evidence that the small sample size results approximate the asymptotic values when the sample is as small as n=20. Similar results are obtained when the estimator in (1) is modified to make it scale invariant.

EXPERIMENTS WITH A PARALLEL COMMUNICATION SCHEME

L.G. VALIANT
Computer Science Department,
Edinburgh University,
Edinburgh, Scotland.

Abstract

The results of simulation experiments to determine the performance of a family of schemes for fast parallel communication are reported.

1. INTRODUCTION

A Parallel Communication Scheme (PCS) consists of a sparse graph and an associated distributed routing algorithm that can realise arbitrary simultaneous communication requests between pairs of nodes of the graph. In a previous paper [1] the author proposed a PCS having the property that all permutation requests could be realised in O(log N) time where N is the number of nodes of the graph. The analysis there, however, does not give good upper bounds on the multiplicative factor in the O(log N) bound. Nor does it extend even in this asymptotic form, to certain attractive related schemes.

Fortunately the algorithm has the curious property (called "testability") of having the same runtime distribution for all permutations. Hence for any value of N its characteristics can be determined by simulating it on just one permutation. We note that probabilities enter only via randomization within the algorithm, and not through any assumed input distributions.

In this paper we report on the results of experiments done to determine just how practical the scheme is for some realistic values of N. By an experiment we mean a set of runs of a computer program from the results of which we can reasonably make deductions about how the program would perform on inputs other than those tried. We distinguish it from the more familiar superstitions run, from which no such deductions can be made.

The statistics we measure are all best regarded as those of freak occurrences. Having sent a large number of packets around a large network we measure how long it takes for the last packet to arrive, the length of the longest queue that ever occurs, and the largest number of packets that ever find themselves at the same node.

After each experiment we could usually find plausible explanations of the phenomena found. However, we were invariably wrong in our predictions about the next experiment.

2. THE ALGORITHMS

All the experiments concerned routing packets among the nodes of an n-dimensional cube. In this graph the node set is $V = \{0,1,\ldots,N-1\}$ where $N=2^n$. For $i \in \{1,\ldots,n\}$ and $x \in V$ x^i denotes the i^{th} most significant bit in the binary representation of x. Also $x//i$ is the number obtained by changing the i^{th} bit of x to its complement. The nN edges of V are therefore the pairs

$$\{(x, x//i) \mid x \in V,\ i \in \{1,\ldots,n\}\}.$$

We ran four different algorithms:

(1) BASIC: essentially as described in [1].
(2) BASIC-FF: BASIC but with a "furthest-to-go first-out" queuing discipline.
(3) 1Q: Similar to BASIC except that there is just one queue at each node.
(4) 1Q-FF: As (3) but with "furthest-to-go first-out" discipline.

In [1] the algorithm was described in such a way that only $O(n)$ bits of book-keeping information needed to be transmitted with each packet. Here we shall follow a slightly simpler presentation that is possible if $O(n\log n)$ bits are allowed.

We label each packet by the address of the node from which it starts. Each of the algorithms consists of two phases A and B. At the beginning of each phase we attach a *ticket* T_x to each packet x. T_x is a vector of length n that describes the precomputed route that the packet is to take. In both phases each packet x executes a procedure of the following form:

```
cobegin  for i:= 1 step 1 until $T_x[i+1] = 0$ or $i = n$
         do Transmit x in dimension $T_x[i]$.
coend
```

Here "Transmit" is a routine that places packet x on to a queue at its current node u waits till x gets to the head of this queue and then transports it to node $u/\!/T_x[i]$ via an edge. The transportation of a packet along a wire takes unit time, and all other computations are assumed to be instantaneous. As there is at most one queue serving any edge only one packet can pass down an edge at any time. In order to specify the algorithms fully we need to (a) name the queues, (b) describe the queuing discipline, and (c) describe how the tickets are prepared initially.

We use the following routines: If D is a set "Pick $d \in D$" means "assigning equal probability to each member of D choose a random element of D and assign it to d". If D,E have equal cardinalities "D:= Perm(E)" means "pick a random $1-1$ correspondence between the elements of D and E, and assign to each element of D the corresponding element of E". If D is an array "Pack (D)" means "if there are r non-zero elements in D assign these to $D(1),\ldots,D(r)$ in order of occurrence and assign zero to $D(r+1)\ldots D(n)$".

BASIC (a) At each node x there are n queues $\{Q(x,i) \mid 1 \le i \le n\}$. $Q(x,i)$ transmits only from node x to node $x/\!/i$.

(b) Standard "first-in first-out" queues. Packets arriving simultaneously are put on to the rear of the queue in order of the labels of the nodes from which they have just been transmitted.

(c) Phase A:

```
begin  $\{T_x[i] \mid 1 \le i \le n\} := \mathrm{Perm}\,\{1,\ldots,n\}$;
  for  i:= 1 step 1 until n do
  begin  Pick $\alpha \in \{0,1\}$;
         If $\alpha = 1$ then $T_x[i] := 0$;
         Pack($T_x$);
  end;
end.
```

```
Phase B  Suppose x is at node u and destined for t;
begin  $\{T_x[i] \mid 1 \le i \le n\} := \mathrm{Perm}\{1,\ldots,n\}$;
  for i:= 1 step 1 until n do if $t^{T_x[i]} = u^{T_x[i]}$
                then  $T_x[i] := 0$;
  Pack($T_x$);
end.
```

BASIC-FF As BASIC except in (b) we have a "furthest-to-go first-out" discipline. At the beginning of each time unit the queue determines its head by scanning its members in order and identifying the first one with a maximal value of "togo". The togo value of a packet x that is at the i^{th} step of its route is $j - i + 1$ where $j = \min \{k | T_x[k] = 0\}$.

1Q (a) At every node x there is just one queue Q(x) which serves all n edges $\{(x, x//i) \mid 1 \le i \le n\}$.

(b) and (c) as BASIC

1Q-FF (a) and (c) as 1Q. (b) as BASIC-FF.

Phase A always starts with one packet at each node addressed to distinct destination nodes. Phase A is independent of the destinations and therefore the distributions of all its statistics must be identical for every such permutation. Phase B is run with input provided by the output of Phase A. For Phase B to be similarly testable we need to have it independent of the start addresses of the packets. Phase A ensures that each packet has been taken to a completely randomly chosen node. To eliminate any possible dependence on the start addresses we need to rearrange in random order the elements of every queue when Phase B is initialized.

In measuring the performance of the algorithms we think of them as synchronized. Just after the start of each time unit one packet is taken from the head of each non-empty queue in the system and is transmitted along the appropriate edge to a neighbouring node. If at the beginning of this time unit $T_x[i+1] = 0$ then the packet x will be considered "finished" when it arrives at the new node at the end of the unit. Otherwise, at the completion of the time unit it will be placed in the appropriate queue at the new node ready for transmission in future time units.

The maximal runtime (mrt) of a run of a phase is the number of time units elapsed before all the packets are finished. The maximal queue length (mql) is the length of the longest queue that ever occurs at any node at any instant between consecutive time units. The maximal node population (mnp) is the largest number of packets (whether finished or in a queue) that ever reside at any one node at any instant between consecutive time units.

3. THE EXPERIMENTS

The simulations were all programmed in PASCAL and run on a DEC VAX 11/780. A linear congruential random number generator supplied by the VAX software was adopted following some experiments, reported in the Appendix, to verify that it was acceptable for the purpose.

The four algorithms BASIC, BASIC-FF, 1Q, and 1Q-FF, were simulated on the identity permutation for values of n up to ten. In addition, Phase B of BASIC was run for $n = 9$ for four distinct permutations which we expected to induce bad behaviour in various senses. The purpose of these latter experiments was to determine whether Phase B on its worst input permutation was better or worse than Phases A and B together. The four "bad" permutations tried are defined below in terms of the mapping of a start address u to its destination address t.

```
Input W  $t^1 \ne u^1$
         for $i \in \{2,3,4,5\}$  $t^i = u^{i+4}$
         for $i \in \{6,7,8,9\}$  $t^i = u^{i-4}$

Input X  $t^1 \ne u^1$
         if $|\{i | t^i = 1$ and $i \ge 2\}| > 4$  then $u^j := t^j$ for $j \in \{2,\ldots,9\}$
         else begin  $u^i := 0$ for $i \in \{2,\ldots,9\}$.
                  for  $i := 2$ step 1 until 9 do if $t^i = 1$ then
                     for $j := i+1$ step 1 until $u^j = t^j = 0$ do $u^j := 1$.
```

[N.B. In the last line $j = 9$ is to be followed by $j = 2$].

Input Y if $|\{i \mid u^i = 1 \text{ and } i \geq 2\}| > 4$ then t is as in W else t is as in X.

Input Z For $i \in \{1,\dots,9\}$ $t^i \neq u^i$.

We note that Input X was suggested by the analysis in Remark 5 of [1].

4. RESULTS

For each value of n between one and ten each of the four algorithms was run one hundred times. Only the results for $n = 9$ and $n = 10$ are presented here. For each value of n we give six tables corresponding to the six statistics measured: maximal runtime, maximal queue length, and maximal node population for each of Phases A and B. The tables allow comparisons among the four algorithms to be made at a glance. Each entry gives the frequency with which the corresponding event occurred in a hundred trials.

	7	8	9	10	11	12	13	14	15	16	17	18
BASIC	-	6	63	29	2	-	-	-	-	-	-	-
BASIC-FF	-	44	56	-	-	-	-	-	-	-	-	-
1Q	-	-	-	-	3	28	43	19	6	1	-	-
1Q-FF	-	-	-	10	51	27	9	2	-	1	-	-

Table 1. Maximal runtime of Phase A $(n = 9)$

	7	8	9	10	11	12	13	14	15	16	17	18
BASIC	-	1	60	37	2	-	-	-	-	-	-	-
BASIC-FF	-	48	52	-	-	-	-	-	-	-	-	-
1Q	-	-	-	-	-	5	29	39	16	6	4	1
1Q-FF	-	-	-	3	17	44	21	10	4	-	1	-

Table 2. Maximal Runtime of Phase B $(n = 9)$

	1	2	3	4	5	6	7	8	9	10	11	12
BASIC	-	19	76	5	-	-	-	-	-	-	-	-
BASIC-FF	-	21	76	3	-	-	-	-	-	-	-	-
1Q	-	-	-	2	39	46	11	2	-	-	-	-
1Q-FF	-	-	-	-	27	42	21	9	1	-	-	-

Table 3. Maximal Queue Length in Phase A $(n = 9)$

	1	2	3	4	5	6	7	8	9	10	11	12
BASIC	-	9	77	14	-	-	-	-	-	-	-	-
BASIC-FF	-	11	84	4	1	-	-	-	-	-	-	-
1Q	-	-	-	-	15	56	20	8	1	-	-	-
1Q-FF	-	-	-	-	7	48	30	12	2	1	-	-

Table 4. Maximal Queue Length in Phase B (n = 9)

	4	5	6	7	8	9	10	11	12	13	14	15
BASIC	-	39	47	14	-	-	-	-	-	-	-	-
BASIC-FF	-	51	39	9	1	-	-	-	-	-	-	-
1Q	-	4	55	34	7	-	-	-	-	-	-	-
1Q-FF	-	7	44	35	10	4	-	-	-	-	-	-

Table 5. Maximal Node Population in Phase A (n = 9)

	4	5	6	7	8	9	10	11	12	13	14	15
BASIC	1	65	29	5	-	-	-	-	-	-	-	-
BASIC-FF	1	64	30	4	1	-	-	-	-	-	-	-
1Q	-	8	44	34	12	1	1	-	-	-	-	-
1Q-FF	-	4	45	32	15	3	-	1	-	-	-	-

Table 6. Maximal Node Population in Phase B (n = 9)

	8	9	10	11	12	13	14	15	16	17	18	19
BASIC	-	5	64	27	4	-	-	-	-	-	-	-
BASIC-FF	-	47	53	-	-	-	-	-	-	-	-	-
1Q	-	-	-	-	-	3	39	41	15	2	-	-
1Q-FF	-	-	-	2	27	45	21	5	-	-	-	-

Table 7. Maximal Runtime of Phase A (n = 10)

	8	9	10	11	12	13	14	15	16	17	18	19
BASIC	-	-	55	40	5	-	-	-	-	-	-	-
BASIC-FF	-	42	58	-	-	-	-	-	-	-	-	-
1Q	-	-	-	-	-	-	7	44	30	12	4	3
1Q-FF	-	-	-	-	8	38	32	14	6	2	-	-

Table 8. Maximal Runtime of Phase B (n = 10)

	1	2	3	4	5	6	7	8	9	10	11	12
BASIC	-	4	89	7	-	-	-	-	-	-	-	-
BASIC-FF	-	6	92	2	-	-	-	-	-	-	-	-
1Q	-	-	-	-	11	50	31	6	2	-	-	-
1Q-FF	-	-	-	-	4	42	43	9	1	1	-	-

Table 9. Maximal Queue Length in Phase A (n = 10)

	1	2	3	4	5	6	7	8	9	10	11	12
BASIC	-	2	92	6	-	-	-	-	-	-	-	-
BASIC-FF	-	-	90	9	1	-	-	-	-	-	-	-
1Q	-	-	-	-	-	29	49	19	2	1	-	-
1Q-FF	-	-	-	-	-	19	53	21	3	4	-	-

Table 10. Maximal Queue Length in Phase B (n = 10)

	4	5	6	7	8	9	10	11	12	13	14	15
BASIC	-	7	60	26	6	1	-	-	-	-	-	-
BASIC-FF	-	15	65	18	1	1	-	-	-	-	-	-
1Q	-	1	25	46	22	6	-	-	-	-	-	-
1Q-FF	-	1	23	51	18	6	1	-	-	-	-	-

Table 11. Maximal Node Population in Phase A (n = 10)

	4	5	6	7	8	9	10	11	12	13	14	15
BASIC	-	23	63	13	1	-	-	-	-	-	-	-
BASIC-FF	-	26	52	19	2	1	-	-	-	-	-	-
1Q	-	-	18	47	26	8	1	-	-	-	-	-
1Q-FF	-	-	14	48	28	6	4	-	-	-	-	-

Table 12. Maximal Node Population in Phase B (n = 10)

The results of a hundred runs of Phase B of BASIC when each of the four candidate "bad" permutations was supplied as input to Phase B directly, are given in Tables 13-15. For comparison purposes the first row gives the corresponding results for a random mapping (generated by a run of Phase A) supplied to Phase B directly.

	7	8	9	10	11	12	13	14	15	16	17	18
Random mapping	-	-	58	33	8	1	-	-	-	-	-	-
Input W	-	-	-	55	38	6	1	-	-	-	-	-
Input X	-	-	-	3	68	24	5	-	-	-	-	-
Input Y	-	-	-	1	65	31	3	-	-	-	-	-
Input Z	-	-	-	-	4	83	12	1	-	-	-	-

Table 13. Maximal runtime of BASIC Phase B for Various Inputs (n = 9)

	1	2	3	4	5	6	7	8	9	10	11	12
Random mapping	-	6	90	4	-	-	-	-	-	-	-	-
Input W	-	-	9	51	32	6	2	-	-	-	-	-
Input X	-	7	66	24	3	-	-	-	-	-	-	-
Input Y	-	-	36	53	9	2	-	-	-	-	-	-
Input Z	-	2	88	10	-	-	-	-	-	-	-	-

Table 14. Maximal Queue Length of BASIC Phase B for Various Inputs (n =9)

	4	5	6	7	8	9	10	11	12	13	14	15
Random mapping	2	58	31	7	2	-	-	-	-	-	-	-
Input W	-	-	-	-	10	36	38	10	5	1	-	-
Input X	-	3	15	36	28	12	5	1	-	-	-	-
Input Y	-	-	3	17	36	23	14	6	-	1	-	-
Input Z	-	50	45	5	-	-	-	-	-	-	-	-

Table 15. Maximal Node Population of BASIC Phase B for Various Inputs(n=9)

Some notable features of Tables 1-12 that were shared by all the runs for $1 \leq n \leq 8$ are the following:

(1) All the distributions considered were sharply peaked. For BASIC and BASIC-FF the variance in all measures was less than one. For 1Q and 1Q-FF it was always less than two.

(2) In BASIC the mean value of mrt was $n + \varepsilon$ where ε increased slowly with n for each phase, from a negative value for $n = 1$ to no more than 0.5 for $n = 10$. The variance in mrt was always less than 0.5 also, and no dependence on n was detected in the range $4 \leq n \leq 10$.

(3) In BASIC-FF $\text{mrt} > n$ never occurred with $n \geq 9$ and only rarely for smaller n.

(4) In both BASIC and BASIC-FF $\text{mnp} > n$ never occurred with $n \geq 9$ and only rarely for smaller n.

(5) The percentages by which the mean statistics for the 1Q algorithms exceeded those for the corresponding BASIC algorithms increased with n. For $n = 10$ these amounted to up to 140% for mql, 50% for mrt and 20% for mnp.

(6) In the runs of Phase B of BASIC on bad inputs the worst we could achieve was degrading mean mrt from 9.5 to 12.1, mean mql from 3.0 to 4.4 and mean mnp from 5.5 to 9.7.

5. CONCLUSIONS

From these preliminary experiments we were pleased to learn the following about the schemes, at least when n is no more than ten.

(a) BASIC performs as well as could be hoped for, requiring only about n steps in each phase. Storage for more than n packets at a node is needed very rarely.

(b) The tail of the runtime distribution above n could be eliminated almost completely by adopting the "furthest-to-go first-out" queuing discipline. This suggests that more complicated adaptive heuristics, where routes may be redirected according to the queue situations encountered in mid-route, are probably not worth considering here.

(c) The 1Q algorithms degrade the performance considerably in every respect, but can be realised, of course, with simpler logic. We note that the best analytic bound we know for their runtime is $O(\log^2 N)$. In this respect the experimental results may be regarded as strikingly positive.

(d) Analysis ([1], Remark 5) shows that for large n on some bad inputs Phase B of BASIC would be far worse than Phases A and B together. We failed to demonstrate this phenomenon for $n = 9$. One could make the following conjecture: for $n = 9$ for all permutations the mean maximal runtime of Phase B is twenty percent less than that of Phases A and B combined. It may be reasonable to proceed in accordance with such a conjecture until a counterexample was found. For some inputs, however, one would then have to pay the price of larger node populations.

6. REFERENCES

[1] L.G. Valiant. A scheme for fast parallel communication. Report CSR-72-80, Edinburgh University Computer Science Department (August 1980)

[2] D.E. Knuth. The Art of Computer Programming, Vol.2, Addison-Wesley (1969).

APPENDIX: CHOOSING A RANDOM NUMBER GENERATOR

In running a simulation experiment we need to convince ourselves that the pseudo-random number generator used behaves reasonably closely to a true random number generator as far as the particular simulation is concerned. The testability of our algorithms suggests the following ad hoc acceptance criterion: Run the algorithm (i.e. phase A followed by Phase B) say 500 times for each of say three different inputs supplied to Phase A. Do this for two distinct pseudo-random number generators. Determine by some statistical test (e.g. chi-square) whether the six sets of results could reasonably have been samples from a common distribution.

As inputs we chose the following: (a) the identity permutation, (b) the complement permutation ($x \to y$ where for each i $x^i \neq y^i$) and (c) $x \to (x+117) \bmod N$.

For random number generators we considered two linear congruential generators. The first one followed the rules-of-thumb recommended by Knuth ([2],p155) except that we chose a small modulus (for low computational complexity) and randomised amongst the algorithms. This Generator I was

$$y_i = (ay_{i-1} + c) \bmod 2^{15}$$

where for each simulation y_0 was a simple function of the current time (in seconds to two decimal places) and the final value of y in the previous simulation. Also $a = 27413 + 8*R$ and $c = 6527 + 2*S$ where R,S are integers in the range [0,99], obtained by calling the generator twice at the beginning of the simulation with $a = 27413$ and $c = 6527$. In general, a random number in the range [0,r) was obtained by computing $\lfloor (y_i*r)/2^{15} \rfloor$

Generator II was a function mth$random provided by the VAX software that produces a real number in the range [0,1). The initial value y_0 was computed for this exactly as in Generator I. A number in the range [0,r) was given by $\lfloor y_i*r \rfloor$.

The BASIC algorithm was simulated 500 times with $n = 8$ for each combination of the three inputs and two generators. The table below gives the results for the runtime for Phase B, which we considered would be one of the more sensitive statistics.

	6	7	8	9	10	11	12	13	14	15	16	17
Inp(a):GenI	-	24	309	154	12	1	-	-	-	-	-	-
Inp(b):GenI	-	23	305	153	17	2	-	-	-	-	-	-
Inp(c):GenI	-	24	324	129	20	3	-	-	-	-	-	-
Inp(a):GenII	-	18	329	131	20	2	-	-	-	-	-	-
Inp(b):GenII	-	17	314	157	12	-	-	-	-	-	-	-
Inp(c):GenII	-	20	318	147	14	1	-	-	-	-	-	-

Table 16. Maximal Runtime of Phase B of BASIC for Various Inputs and Generators.

Because a runtime of eleven occurred with very small frequency (< 5) in every case, we combined these occurrences with the previous column. Each run can therefore have four different outcomes. This gives a value of $\chi^2 = 15.7$ which is near the sixty percentile point in the χ^2 distribution with fifteen degrees of freedom. On this basis we considered that both generators had passed the acceptance test for simulating the run-time of Phase B of BASIC for $n = 8$.

Strictly speaking we ought to have tried the above six combinations of input and generator for each of the testable properties discussed in this paper. In the event we ran the experiments consistently only for Generator II and input(a), trying other combinations only occasionally as spot checks.

A PROCESSOR-EFFICIENT CONNECTED COMPONENTS ALGORITHM

C.M. FIDUCCIA and R.M. MATTHEYSES
General Electric
Research and Development Center
Schenectady, NY 12345

ABSTRACT

The connected components of an undirected graph of n vertices and e edges can be computed in $O(\log^2 n)$ time using $O(e)$ processors. These bounds are achieved on a parallel computer in which processors have access to a common memory. Simultaneous access to a memory location is allowed for reading but simultaneous writes or writes and reads are not permitted.

INTRODUCTION

The connected components of an undirected graph $G=(V,E)$ having $n=|V|$ vertices and $e=|E|$ edges can be computed in time $O(e)$ or $O(n^2)$ depending on whether G is represented by adjacency lists or an adjacency matrix, respectively. Hirschberg, et al [1] have shown that on a parallel computer, in which processors have access to a common memory, the connected components of G can be computed in $\log^2 n$ time using $n^2/\log n$ processors. Because they assume that G is presented as an adjacency matrix, at least $\Omega(n^2)$ work is required to determine whether or not an edge is present. This implies that at least $\Omega(n^2/\log^2 n)$ processors are required, since the processor-time product must be at least as large as the work required. In this paper, we assume that the graph G is presented as a set of edges (vertex pairs). We then give an algorithm which solves the connected components problem in $\log^2 n$ time using max(n,e) processors. The algorithm can be used to compute the reflexive transitive closure R^* of a symmetric relation R, provided R is given as a set of pairs and R^* is specified by its equivalence classes, rather than by nxn bit matrices.

THE ALGORITHM CONNECT

Our algorithm is a refinement of the algorithm CONNECT given by Hirschberg, et al [1]. Their algorithm assumes that the undirected graph G=(V,E) has a vertex set V={0,1,...,n-1}.

```
Algorithm CONNECT:
Input: The nxn adjacency matrix A[0:n-1,0:n-1] of an
       undirected graph.
Output: The vector D[0:n-1] such that D[i] equals the
        smallest numbered vertex in the connected
        component to which node i belongs.
Comment: Each of the following steps is executed in
         parallel for all i,0≤i<n. The assignments
         in the various steps are done simultaneously
         for all i.

  1. for all i do D[i]←i
     do steps 2 through 6 for ⌈logn⌉ iterations
  2.    for all i do C[i]←min{D[j]|A[i,j]=1 and D[j]≠D[i]}
                            j
                                   if none then D[i]
  3.    for all i do C[i]←min{C[j]|D[j]=i and C[j]≠i}
                            j
                                   if none then D[i]
  4.    for all i do D[i]←C[i]
  5.    for ⌈logn⌉ iterations do
            for all i do C[i]←C[C[i]]
  6.    for all i do D[i]←min(D[i],C[i])
```

We see that steps 1,4,5 and 6 can be done with n processors, while steps 2 and 3 appear to require n^2 processors. This is because we need to compute n minima and each minimum can be computed in logn time with n processors. In fact a minimum can be computed in 2logn time with only n/logn processors; so that n^2/logn processors suffice while maintaining an $O(\log^2 n)$ running time.

Our new algorithm is made processor-efficient by using only max(n,2e) processors to perform the minimum computations in steps 2 and 3. All other steps of CONNECT remain the same, except for the fact that the input is no longer an nxn adjacency matrix A but rather, an ex2 matrix E representing the set of e undirected edges of the graph. No particular order is assumed for the edges in E; however, multiple edges are forbidden.

SOLVING THE CONTENTION PROBLEM

Let us focus on the min computations in steps 2 and 3 of CONNECT for a typical vertex i, say i=1. We thus need to perform

$$C[1] \leftarrow \min_{j}\{D[j] \mid (1,j) \varepsilon E \text{ and } D[j] \neq D[1]\}$$

and

$$C[1] \leftarrow \min_{j}\{D[j] \mid D[j]=1 \text{ and } D[j] \neq C[j]\}$$

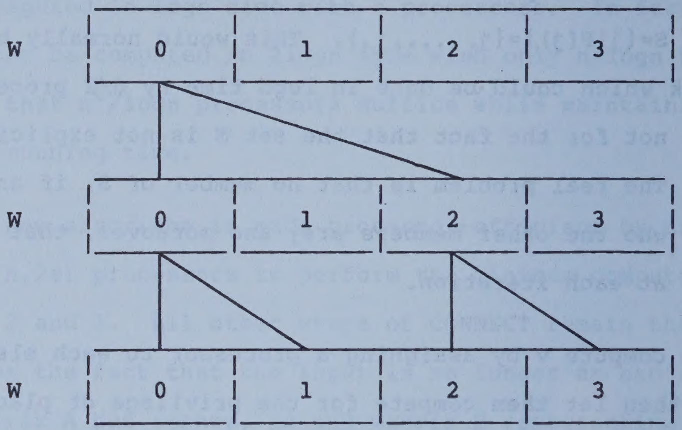
Note that both of these are of the form

$$v \leftarrow \min_j \{f(j) \mid P(j)\}$$

for some function f over V and some predicate P on V. We will compute v with d processors, where $d=|S|$ is the cardinality of the set $S=\{j \mid P(j)\}=\{j_1,\ldots,j_d\}$. This would normally be a simple task which could be done in logd time by d/2 processors, were it not for the fact that the set S is not explicitly known. The real problem is that no member of S, if any, "knows" who the other members are; and moreover, that S changes at each iteration.

We compute v by assigning a processor to each element of S. We then let them compete for the privilege of placing their f values into v. At most one processor will be successful. This is accomplished by using an array W[0:n-1] as the "battlefield". We cannot afford to initialize W, since we have only d processors available. The scheme we will use is similar to the one used by Hirschberg in [2]. Processors $p_1,\ldots,p_d$ initially "occupy" locations $W[j_1],\ldots,W[j_d]$, carrying their respective f values $f(j_1),\ldots,f(j_d)$. Then in a binary-tree fashion, they will compete with other processors in the array W. If two processors confront each other, the winner will proceed with the min of their f values, while the loser expires. This scenario is repeated logn times until the sole survivor, if any, resides at location W[0] with the value $v=\min\{f(j) \mid j \in S\}$. Note that the min could be replaced by any associative binary operator.

The contention scheme is easily visualized with the aid of the diagram below, where $n=4$.



For example, the processors (if any) located at W[2] and W[3] confront each other and the winner moves to location W[2]. This particular scheme has the property that at the k^{th} stage of contention, $k=0,\ldots,\lceil \log n\rceil-1$, the adversary of the processor at location $j=0,\ldots,n-1$ is at location $j'=j \oplus 2^k$, i.e. j and j' differ only in their k^{th} bit. Thus for $k=0$ we have $2 \oplus 2^k=3$ and $3 \oplus 2^k=2$, while for $k=1$ we have $0 \oplus 2^k=2$ and $2 \oplus 2^k=0$.

The contention scheme for placing $\min\{f(j)\,|\,j\ S\}$ in W[0], is given below. It uses $d=|S|$ processors and takes $O(\log n)$ time. Note that each encounter consists of determining the minimum f value regardless of the presence of an adversary, detecting the presence or absence of an adversary, and deciding who survives. The survivor continues on with the next stage of the minimization process. The array W is in common memory; all other variables are local to each of the d processors. The constants "absent" and "present" are any two distinct values.

```
for all j in S do
    position←j
    value   ←f(j)

    for k←0 to  ⌈log|V|⌉-1  do
        adversary    ←position ⊕ 2^k
        W[adversary]←value
        W[position] ←value
        value        ←min(value,W[adversary])

        W[adversary]←absent
        W[position] ←present
        if W[adversary]=present and adversary < position
           then expire

        position    ←min(position,adversary)
```

The above scheme contains a minor error. If n is not a power of two, the value computed for adversary may not be a valid index of the array W[0:n-1]. For example, when n=5, k=0 and j=4, we get the invalid index $4 \oplus 1=5$. This problem is solved by simply placing all statements below the assignment

$$\text{adversary} \leftarrow \text{position} \oplus 2^k$$

in the body of the conditional statement

if adversary < |V| then do.

This modification is included in the implementation of steps 2 and 3 of CONNECT given in the next section. If the binary operator being evaluated during the battle is also commutative, the adversary calculation can be simplified by relating the first and last elements, the second and last - 1 and so on. That is, we match the elements that are i in from each end of the battlefield. After the battle the victors will be in the front half of the vector. Thus we can replace the adversary calculation with

$$\text{adversary} \leftarrow \lceil n/2^k \rceil - (\text{position}+1)$$

where k is the number of the current battle.

THE NEW ALGORITHM

Having given the contention scheme for computing one min, in a general setting, we now tailor it to the specific needs of realizing steps 2 and 3 of the algorithm CONNECT. We will show that steps 2 and 3 can be performed in logn time using max(n,2e) and n processors respectively. In both cases, we need to simultaneously compute n new values C[0],...,C[n-1]. We do this by executing n independent contention algorithms, i.e. one for each i. Because this requires n copies of the W array, we expand W into an nxn array W[0:n-1,0:n-1]; the i^{th} row being the battleground for the computation of C[i].

Recall that the graph G is given as an ex2 array E[1:e,1:2] in which an edge between vertices r and s is represented by setting E[k,1]=r and E[k,2]=s, for some k. In order to perform step 2, we assign two processors to each edge of G. This is possible because we know explicitly where the edges are; namely, in the array E, as opposed to having to search for them were G to be given as an adjacency matrix A. The processor assigned to each E[k,1] will carry the value D[r] in the contention scheme which updates C[s], while the processor assigned to E[k,2] will in a similar manner, carry the value D[s] for updating C[r]. The resulting algorithm segment is given below. The notation {i,j} E is understood to mean that the edge between vertices i and j gives rise to the two ordered pairs (i,j) and (j,i).

```
Algorithm for step 2 of CONNECT:
for all i in V do W[i,0]←D[i]
for all {i,j} in E do
    position←j
    value   ←D[j]
    for k←0 to $\lceil \log|V| \rceil - 1$ do
        adversary←position $\oplus 2^k$
        if adversary < n then do
            W[i,adversary]←value
            W[i,position] ←value
            value         ←min(value,W[adversary])
            W[i,adversary]←absent
            W[i,position] ←present
            if W[i,adversary]=present and
               adversary < position
               then expire
            position←min(position,adversary)
for all i in V do C[i]←W[i,0]
```

As written, the above algorithm uses 2e processors for the "for all {i,j} in E" segment. It is clear that by duplicating the code, we can use only e processors for this segment at the expense of doubling the run time.

In order to perform step 3, we use only n processors. This is because the condition D[j]=i in step 3 shows that the value D[j] contends to update only one of the C values. Thus we can assign a processor to each vertex j. The processor looks at the value of D[j] and enters the contention for updating location C[i]=C[D[j]].

Algorithm for step 3 of CONNECT:

```
for all i in V do W[i,0]←D[i]

for all j in V do

    if D[j]=C[j] then expire
    position←j
    value   ←C[j]
    i       ←D[j]

    for k←0 to ⌈log|V|⌉-1  do

        adversary←position ⊕ 2^k

        if adversary < n then do

            W[i,adversary]←value
            W[i,position] ←value
            value         ←min(value,W[i,adversary])

            W[i,adversary]←absent
            W[i,position] ←present
            if W[i,adversary]=present and
                adversary < position
                then expire

            position      ←min(position,adversary)

for all i in V do C[i]←W[i,0]
```

Using the above algorithms for steps 2 and 3 of CONNECT gives a new connected components algorithm which uses O(e) processors and runs in $O(\log^2 n)$ time. It is worth noting that the reduction in processors is not accompanied by a reduction in the work space required by the algorithm. $O(n^2)$ space (the array W) is still required to solve the contention problem. In spite of the fact that at most elogn locations of W are ever accessed, the entire n^2 locations appear necessary since we don't know a priori which of them will in fact be used by the algorithm. If we assume that the input graph G is given in a more structured form, it is possible to perform step 2 of CONNECT in O(e) space. Thus the problem reduces to doing step 3 with O(e) space. This is rather odd, since there are never more than n processors competing in W during step 3; so that we never use more than nlogn locations of W. On the other hand, step 3 is a very dynamic step, because the positions of the processors are determined by the D[j] values; and these change at each iteration.

REFERENCES

1. Hirschberg, D.S., Chandra, A.K. and Sarwate, D.V., Computing Connected Components on Parallel Computers. Comm. ACM 22,8 (Aug. 1979), pp. 461-464.

2. Hirschberg, D.S. Fast Parallel Sorting Algorithms. Comm. ACM 21,8 (Aug. 1978), pp. 657-661.

LINEAR TRANSFORMATION PERMUTATIONS IN MULTISTAGE INTERCONNECTION NETWORKS*

MOHAMMAD A. ABIDI AND JOHN J. METZNER
Wayne State University, Detroit, Michigan

SUMMARY

Consider an n-stage interconnection network employing 2x2 switching elements, having $N=2^n$ inputs and $\bar{N}$ outputs. Define an LT permutation as a one-to-one linear mapping: $\bar{d}=M\bar{s}$, where $\bar{d}$ and $\bar{s}$ are nx1 binary n-tuples representing the destination and source addresses, respectively. Simple rules relating to the rank of certain submatrices of the transformation matrix are found to prescribe which of these permutations can be passed conflict-free by each of various multi-stage interconnection networks.

Define Q_N as the number of LT permutations which can be passed conflict-free by a given network, and define R_N as the total number of LT permutations. Then it can be shown that

$$Q_N = N^{\log_2 N-1} \qquad \text{and} \qquad Q_N/R_N \cong 1/.289\ N$$

Let P_N be the total number of conflict-free permutations (all kinds), and let T_N be the total of all permutations.

$$P_N/T_N \cong \frac{1}{\sqrt{2\pi N}}\left(\frac{e^2}{N}\right)^{N/2}$$

It can be seen that for moderately large N the fraction of LT permutations which are conflict-free (Q_N/R_N) is very much greater for these networks than the probability that an arbitrarily chosen permutation is conflict-free (P_N/T_N). Thus, LT permutations are a rich source of conflict-free permutations. The number of conflict-free LT permutations is the same for all networks considered, but the conflict-free sets are different for different networks.

We can obtain a larger conflict-free family of permutations for a given network by considering pseudo-LT permutations of the form $\bar{d}=M\bar{s}\oplus\bar{a}$, where $\bar{a}$ is an arbitrary but fixed binary vector. The permutation $\bar{d}=M\bar{s}+\bar{a}$ can be shown to be conflict-free iff $\bar{d}=M\bar{s}$ is conflict-free for the network. This increases the number of classified conflict-free permutations by a factor N, and yields a number of conflict-free pseudo LT permutations U_N as

$$U_N = NQ_N = N^{\log_2 N}$$

Compared to the total number of conflict-free permutations for a given network, P_N, we find that

$$U_N/P_N = N^{(n-N/2)}$$

---

*This material is based upon work partially supported by the National Science Foundation under Grant ENG 75-07839. It is part of a dissertation submitted by M.A. Abidi to Wayne State University in partial fulfillment of the requirement for the Ph.D. degree.

ELECTION PROCESSES IN DISTRIBUTED SYSTEMS

D.S. HIRSCHBERG
Rice University, Houston TX

INTRODUCTION

We are given n processors that work asynchronously. Each of the processors has an associated unique value and none of the processors has a priori knowledge of the number of processors in the network. The problem is to designate by consensus a unique processor from the system. Any one or several of the processors may independently initiate such an election. The total number of data transmissions (messages passed) among the n processors is a measure of the complexity of a solution algorithm.

LeLann [IFIP 77, 155-160] presented an algorithm that requires $O(n^2)$ messages in the case when the processors are arranged in a circle. Chang and Roberts [CACM 22 (5/79), 281-283] proposed an improved algorithm that requires only $O(n \log n)$ messages on the average but, in the worst case, still requires $O(n^2)$ messages. Hirschberg and Sinclair [CACM 23 (11/80), to appear] exhibit an algorithm that requires $O(n \log n)$ messages in the worst case. Burns [Indiana U. CS Dept. TR-91, May 1980] proved that, for the circular arrangement, $n \log n$ is asymptotically optimal.

AN ELECTION ALGORITHM FOR TREES

We first consider the case in which the processors are arranged acyclically. We assume that messages can be transmitted bidirectionally on any transmission line. We propose an algorithm that requires $O(n)$ messages in the worst case.

The algorithm can be described as follows. News that an election is to be held is propagated in all directions. The leaves of the tree send their values to the next level. Each interior vertex of the tree, all but one of whose neighbors being leaves, now is aware of the local best value of all nodes toward the exterior of the tree and passes this information up the tree. This process of passing the best value of all lower nodes toward the interior of the tree continues, albeit asynchronously, until a node receives information from all adjacent nodes thus having global information. The winner being determined, this news is propagated to the rest of the tree.

AN ELECTION ALGORITHM FOR ARBITRARY GRAPHS

We now describe an algorithm for the case when cycles may be present. The total number of messages passed during the entire election process is no more than $2(n+1)e$, where e is the number of edges in the network.

A node y initiates its own candidacy by asking its neighbors to "ELECT y". When x is presented by such a request, x can veto it if (1) $x>y$ (and x, in turn, would run for election) or if (2) x learns from asking x′s neighbors to "ELECT y" that y is not best.

The procedure of requests to elect y imposes a spanning tree upon the network. An edge (x,z) of the spanning tree for y′s election is defined by node x first learning from z that y is running for election. Information that y can or cannot win is transmitted along edges of this tree.

ON PARALLEL SEARCHING OF FILES IN A DISTRIBUTED COMPUTING SYSTEM*

S. L. DALLAS AND S. LOUIS HAKIMI
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, Illinois 60201

Abstract: Ghosh suggested techniques to minimize retrieval times in a distributed computing system by distributing the files such that each file of a given query (access request) may be searched in parallel. Creating such a file distribution is reconsidered using graph theoretic constructs. Results are presented that demonstrate the general problem, as well as several variations, are NP-hard.

INTRODUCTION

A distributed computer system (briefly, a distributed system) is an interconnection of processing elements which can communicate with one another. Each processing element is called a node; these nodes may be "back-end" processors [4], associated memories and processors [16], or general purpose computers. A node may function independently of the other nodes in a distributed system.

An access request or a "query" to a distributed system may arise at any of the nodes. Furthermore, the requested data may have been allocated to any node in the system. Clearly, the retrieval time for any query is dependent on the data transmission (or communication) rate and on the amount of time needed to search the required files. This leads us to the file allocation problem in distributed systems which may be described as follows. Given n files and d nodes in a distributed system, it is required to allocate the files to the nodes while minimizing some cost function. Eswaran [6] has shown that the general file allocation problem is NP-hard [11,12].

Ghosh [9] considered the possibility of creating a file allocation scheme which reduces retrieval time by allowing for parallel searching of multiple (distinct) files. This paper reconsiders and generalizes the preliminary work of Ghosh by reformulating the problem as a graph coloring problem. Extensions and variations of the original problem are also presented. Note that we concentrate on parallel searching capability and issues such as communication cost are not considered.

THE CPS PROBLEM

Before stating Ghosh's problem, we delineate the assumptions that need to be made. The problems considered in this paper deal only with access requests (retrieval) of stored data; that is, operations such as insertion,

---

*This work was supported by the National Science Foundation under Grant ECS-7909724.

deletion, etc. are not considered. We assume that a file is the basic unit of allocation in a distributed system; that is, all occurrences of a given record type must be stored together. Each query to the distributed system requests a subset of the set of files in the distributed system. Consequently, each file in this subset must be accessed to satisfy the query. Ghosh reasoned that the retrieval time for a given query is minimized if each file in the set of requested files could be searched in parallel with the other files in the query. This parallel searching could be accomplished if and only if the files requested by each query are allocated to different nodes in the distributed system. Such a file allocation scheme is said to be Completely Parallel Searchable (CPS).

The CPS problem may be stated more precisely as follows. Let $S = \{s_1,s_2,\ldots,s_n\}$ be the files in a distributed system. The queries to the system form a set $Q = \{q_1,q_2,\ldots,q_k\}$. Each query, q_i, requires that a number of files in S be searched. If files $s_{i_1},s_{i_2},\ldots,s_{i_m}$ are requested by query q_i, we denote this by $q_i = \{s_{i_1},s_{i_2},\ldots,s_{i_m}\}$. Therefore, Q represents a collection of subsets, $q_1,q_2,\ldots,q_k$, of S. We assume that $S = \bigcup_{i=1}^{k} q_i$, i.e. each file is requested by at least one query. The CPS problem is: find (if possible) a partition of S into ℓ subsets $B_1,B_2,\ldots,B_\ell$, (that is, $B_i \cap B_j = \emptyset$ for $i \neq j$ and $\bigcup_{i=1}^{\ell} B_i = S$), such that if s_a and $s_b \in q_i$ and $s_a \in B_j$, then $s_b \notin B_j$. Equivalently, the problem may be stated as follows: for some integer ℓ find a partition of S into subsets $B_1,B_2,\ldots,B_\ell$ such that $|B_j \cap q_i| \leq 1$ for all i and j, $1 \leq i \leq k$ and $1 \leq j \leq \ell$; where $|X|$ represents the cardinality of set X. We note that if the files in each subset, B_j, $1 \leq j \leq \ell$, are assigned to distinct nodes in the distributed system, then the complete parallel searching of all queries is ensured. We also observe that the resulting file allocation scheme is non-redundant; that is, only one copy of each file exists in the distributed system.

The CPS problem can be easily solved if ℓ is allowed to be sufficiently large, say $\ell = n$. The real problem is to minimize ℓ such that the system is CPS; i.e. find the least number of nodes needed to allow for parallel searching in every query of Q. This problem will be called the minimum CPS problem. Ghosh proposed an algorithm to solve the CPS problem. However, his algorithm does not solve the minimum CPS problem. In this paper, we will concentrate on the minimum CPS problem. It will be seen that our approach also immediately leads to better algorithms (than Ghosh's) for the CPS problem.

Given sets S and Q, we construct a graph G(Q) to be called the query graph, whose vertex set and edge set, denoted by V(G(Q)) and E(G(Q)), are as follows: $V(G(Q)) = S$ and $E(G(Q)) = \{(s_i,s_j) \mid$ there exists $q_\alpha \in Q$ such that s_i and $s_j \in q_\alpha\}$. Figure 1 shows the query graph for queries $Q = \{\{s_1,s_2,s_3\}, \{s_3,s_4\}, \{s_1,s_5,s_6\}, \{s_2,s_3,s_5\}\}$.

Given a graph G, a coloring of G is an association of colors to the vertices of G such that adjacent vertices have different colors. The minimum number of colors used among all possible colorings of G is called the chromatic number $\chi(G)$ of G.

Let G(Q) be the query graph of query set Q and a set S of files. Then, it is easy to see that any coloring of G(Q) corresponds to a solution of the underlying CPS problem and vice-versa. Similarly, finding a coloring with $\chi(G(Q))$ colors corresponds to a solution of the minimum CPS problem. As any graph can be the query graph for some set of queries and files, we have the following result.

Theorem 1: The minimum CPS problem is NP-hard.

Having formulated the CPS problem as a graph coloring problem, one can apply known results in graph coloring to obtain results for the CPS problem. To begin with, one can use bounds on $\chi(G)$ to obtain upper and lower bounds on ℓ which are tighter than the bounds given by Ghosh. For example, if G contains a clique (complete subgraph) with c vertices, then $\chi(G) \geq c$. (Admittedly, finding the largest such c is NP-hard [8], nevertheless this result often yields a tighter lower bound than the one suggested by Ghosh. Also, $\chi(G)$ can be arbitrarily large even if the maximum c is equal to 2 [5].) On the other hand, the bound which was first presented by Brooks [3] yields a good general upper bound on $\chi(G)$. We now describe another upper bound on $\chi(G)$ due to Welsh and Powell [17] which yields not only a tighter bound than Brooks' Theorem but the proof of this result also gives an algorithm to find a coloring. The degree of vertex v_i, denoted by d_i, is the number of edges of G incident at v_i. Without loss of generality, assume the n vertices of G are ordered such that $d_1 \geq d_2 \geq \ldots \geq d_n$.

Theorem 2 [17]: $\chi(G) \leq \max\limits_{1 \leq i \leq n} \min \{d_i+1, i\}$.

Proof: The colors are identified by integers 1,2,.... Consider the following algorithm for coloring the vertices of G. Color vertex v_1 by color 1. Proceed by sequentially assigning the least numbered color to vertex v_i such that adjacent vertices have different colors. It is clear that the color assigned to $v_i \leq i$. Also as the degree of v_i is d_i, the color assigned to $v_i \leq d_i + 1$. Since we wish to produce a coloring using the least number of colors, the theorem immediately follows.

The above proof provides us with a coloring algorithm, but there are other efficient coloring algorithms [2,15] which, on the average, color the graph with fewer colors; in fact, these algorithms often color a graph with exactly $\chi(G)$ colors.

If the query graph G has special properties, say it is planar (resp., bipartite) then, of course, we know $\chi(G) \leq 4$, (resp., $\chi(G) = 2$), [1,10].

In summary, various graph coloring results can be used either (1) to obtain bounds on the required number of parts ℓ in a partition of the set of files S, or (2) to color the query graph which in turn yields a CPS allocation scheme for the files, if the number of colors used does not exceed the number of nodes in the distributed system.

CPS FILE ALLOCATION OVER A SMALL NUMBER OF NODES

Suppose a solution to the minimum CPS problem has been found. Let ℓ be the number of parts in the resulting partition of the set S of files. This implies that at least ℓ nodes must be present in the distributed system to create a CPS file allocation scheme. Suppose that the number of nodes in the distributed system is d, and $d < \ell$. We shall examine this problem when file redundancy is, or is not, allowed.

Case 1: If file redundancy is not allowed, then there is exactly one copy of each file in the distributed system. Under this assumption, in any file allocation scheme, there will be some queries whose files cannot be searched in parallel. It is necessary, therefore, to modify the CPS problem such that the solution now aims at achieving "maximum parallelism". We will describe a quantitative measure of the degree of parallelism which is closely related to the average time required to search the files of any query.

We begin by assuming that all queries are equally likely. Let Π be the family of all partitions of S into d parts. Let $\pi = \{B_1, B_2, \ldots, B_d\} \in \Pi$. (Observe that some B_i's may be empty). Consider query $q_i \in Q$. Define $t(q_i,\pi) = \max_{1 \leq j \leq d} |q_i \cap B_j|$. We note that $t(q_i,\pi)$ is proportional to the retrieval time for query q_i, given the file partition π. Then $T(Q,\pi) = \sum_{i=1}^{|Q|} t(q_i,\pi)$ is proportional to the average retrieval time, given the file partition π. We wish to find an (optimum) partition $\pi_0 \in \Pi$ such that $T(Q,\pi_0) \leq T(Q,\pi)$ for all $\pi \in \Pi$. We plan to demonstrate that the goal of finding an optimum partition is unrealistic and to propose possible heuristic techniques for finding a near optimum partition.

Given a set of files S and a set of queries Q, we define a weighted query graph $G_w(Q)$ to be the query graph G(Q) with the weight of each edge $(s_i, s_j) \in E(G(Q))$ being equal to the number of queries which require both files s_i and s_j.

Theorem 3: The problem of finding an optimum partition is NP-hard even if $|q_i| = 2$ for $i = 1,2,\ldots,|Q|$, $d = 2$, and no two queries request the same files.

Proof: Since $d = 2$ and $|q_i| = 2$ for $i = 1,2,\ldots,|Q|$, it can be seen that an optimum partition is a partition of files into exactly two parts such that the maximum number of queries in Q are completely parallel searchable. Consider the weighted query graph $G_w(Q)$. We note that the weight of each edge is equal to 1 since each edge, say (s_i,s_j), corresponds to a unique query, say $q_k = \{s_i,s_j\}$ and each partition $\pi = \{B_1,B_2\}$ corresponds to an edge-cut $E(B_1,B_2)$ of $G(Q)$, where $E(B_1,B_2) = \{(s_i,s_j) \in E(G(Q)) | s_i \in B_1$ and $s_j \in B_2\}$. Query $q_k = \{s_i,s_j\}$ is completely parallel searchable with respect to partition $\pi = \{B_1,B_2\}$ if and only if edge (s_i,s_j) belongs to the edge-cut $E(B_1,B_2)$. Thus, the problem of finding an optimum partition is equivalent to that of finding a maximum edge-cut in $G(Q)$ [7]. As $G(Q)$ is an arbitrary graph, this immediately proves the theorem.

Even the problem of developing heuristic algorithms which yield a near optimum partition of files is quite difficult. One possible approach is as follows. Let $G(V,E)$ be a weighted graph. Let $B_1,B_2,\ldots,B_d$ be a partition of the vertices of G with $B_i \neq \emptyset$. The partition $B_1,B_2,\ldots,B_d$ produces a d-cut of G; namely, $E(B_1,B_2,\ldots,B_d) = \{(v_i,v_j) \in E \mid \text{if } v_i \in B_k$ for some k, then $v_j \notin B_k\}$. A maximum d-cut is a d-cut of G for which the sum of the weights of its edges is maximum. Kernighan and Lin [13] have proposed a heuristic algorithm for finding a maximum d-cut.

It is expected that the problem of finding an optimum partition is closely related (but not equivalent) to that of finding a maximum d-cut of the weighted query graph $G_w(Q)$ and thus the heuristic procedure of Kernighan and Lin could be quite helpful in finding near optimum results.

Case 2: If file redundancy is allowed, then there may exist more than one copy of certain files in the distributed system. Under this assumption, one can often achieve complete parallel searching capability. Our goal is to achieve the CPS property with "minimum file redundancy". Let $\beta_i \subset S$ be the set of files allocated to node i of the distributed system, $i = 1,2,\ldots,d$. We naturally assume that $\bigcup_{i=1}^{d} \beta_i = S$, but here $\beta_i \cap \beta_j$ is not necessarily empty. We call $\{\beta_1,\beta_2,\ldots,\beta_d\}$ a file allocation scheme with redundancy. Let $N_j(\beta_1,\beta_2,\ldots,\beta_d)$ be the number of times file s_j appears in the system given the allocation $\{\beta_1,\beta_2,\ldots,\beta_d\}$, $j = 1,2,\ldots,|S|$. Let $R(\beta_1,\beta_2,\ldots,\beta_d) = \sum_{i=1}^{|S|} N_i(\beta_1,\beta_2,\ldots,\beta_d) - |S|$. We use $R(\beta_1,\beta_2,\ldots,\beta_d)$ as the measure of file redundancy given the allocation scheme $\{\beta_1,\beta_2,\ldots,\beta_d\}$. The problem is to find (if possible) a file allocation $\{\beta_1^*,\beta_2^*,\ldots,\beta_d^*\}$

which achieves the CPS property (for the given query set Q) such that $R(\beta_1^*,\beta_2^*,\ldots,\beta_d^*) \leq R(\beta_1,\beta_2,\ldots,\beta_d)$ for all possible CPS file allocations $\{\beta_1,\beta_2,\ldots,\beta_d\}$ with redundancy. We call $\{\beta_1^*,\beta_2^*,\ldots,\beta_d^*\}$ an optimum CPS file allocation. We begin with the following simple result. The proof is left to the reader.

Lemma 1: Given S and Q, an optimum CPS file allocation scheme (with redundancy) exists if and only if $d \geq |q_i|$ for $i = 1,2,\ldots,|Q|$.

We now attempt to relate the problem of finding an optimum CPS file allocation scheme with redundancy to a variation of the graph coloring problem, and then to prove that the problem is NP-hard. We begin by an example. Consider the files $S = \{s_1,s_2,s_3,s_4,s_5\}$ and queries $Q = \{\{s_1,s_2\}, \{s_1,s_3\}, \{s_1,s_4\}, \{s_1,s_5\}, \{s_2,s_3\}, \{s_3,s_4\}, \{s_4,s_5\}, \{s_5,s_2\}\}$. The query graph G is shown in Fig. 2a. We note that $\chi(G) = 3$. Thus, if $d = 2$, one cannot have a CPS file allocation scheme without redundancy. However, if redundancy is allowed, then one can have two copies s_1' and s_1'' of file s_1 and replace queries $\{s_1,s_2\}$ and $\{s_1,s_4\}$ by $\{s_1',s_2\}$ and $\{s_1',s_4\}$, and queries $\{s_1,s_3\}$ and $\{s_1,s_5\}$ by $\{s_1'',s_3\}$ and $\{s_1'',s_5\}$. Having introduced this redundancy, the resulting query graph G' is shown in Fig. 2b. We note that $\chi(G') = 2$ and consequently the following file allocation scheme $\{\{s_1',s_3,s_5\}$ $\{s_1'',s_2,s_4\}\}$ is CPS.

To simplify our discussion, we will assume that $|q_i| = 2$ for $i = 1,2, \ldots,|Q|$. Let $G(V,E)$ be a graph. Graph G' is called an elementary expansion of G, if G' is obtained from G by taking a vertex $v \in V$ and replacing it by two vertices v' and v'' in G' and by replacing each edge incident at v in G, say (v,u), by either (v',u) or (v'',u) in G'. The remaining vertices and edges of G' are the same as the corresponding ones in G. Note that G' has one more vertex than G and has the same number of edges as G. Graph G_1 is called an expansion of G if G_1 is obtained from G by a sequence of elementary expansions. The order of this expansion is $|V(G_1)| - |V(G)|$. Now, it can be seen that the problem of finding an optimum CPS file allocation scheme with redundancy, when $|q_i| = 2$ for $i = 1,2,\ldots,|Q|$, can be formulated as follows.

Given files S and queries Q with $|q_i| = 2$ for $i = 1,2,\ldots,|Q|$, the number of nodes d in the distributed system, and assuming that $\chi(G(Q)) > d$, we wish to find a minimum order expansion G_1 of $G(Q)$ such that $\chi(G_1) = d$.

Theorem 4: The problem of finding an optimum CPS file allocation scheme with redundancy is NP-hard even if $|q_i|=2$ for $i = 1,2,\ldots,|Q|$ and $d = 2$.

Proof: As our previous discussion indicates, this problem is equivalent to the following problem: given an arbitrary graph G with $\chi(G) > 2$, find a minimum order expansion G_1 of G such that $\chi(G_1) = 2$. It can be easily seen that in a minimum order expansion of G no vertex will have more than two copies. (Intuitively, one need not have any more than d copies of any file, say one at each node.) Let G_1 be a minimum order expansion of G. Let $V_1 \subset V(G)$ be the vertices of G each of which has exactly two copies in G_1. We claim that G_1 is a minimum order expansion of G if and only if V_1 is a smallest set of vertices of G such that $\chi(G-V_1) = 2$, where $G-V_1$ is the subgraph of G "induced" on $V-V_1$ (i.e., the subgraph of G that remains after the vertices in V_1 are deleted). To prove this claim, we proceed as follows.

Let G_1 be a minimum order expansion of G and $V_1 \subset V$ be the vertices which have two "representatives" in G_1. Then, it is clear that $\chi(G-V_1)=2$. Suppose now that there exists a set $V_2 \subset V$ such that $|V_2| < |V_1|$ and $\chi(G-V_2) = 2$. We plan to show that G_1 could not be a minimum expansion of G such that $\chi(G_1) = 2$. To do this, we color the vertices of $G-V_2$ by two colors, say $\{0,1\}$, and proceed by providing two representatives for each vertex in V_2 and color these two vertices by the two colors. Let V_2^k be the representatives of the vertices in V_2 having color k (k=0,1). We construct graph G_2 as follows: $V(G_2) = (V-V_2) \cup V_2^0 \cup V_2^1$ and each edge $(u,v) \in E(G)$ becomes an edge in G_2 between representatives of vertices u and v having different colors. We now observe that G_2 is an expansion of G which is of lower order than G_1.

On the other hand, if V_1 is a smallest set such that $\chi(G-V_1) = 2$, then by the above procedure one can construct an expansion of G' of G such that $\chi(G') = 2$. Furthermore G' is a lowest order expansion of G, because otherwise one can easily produce a set of vertices V_2 with $|V_2| < |V_1|$ and $\chi(G-V_2) = 2$.

The problem of finding a smallest subset V_1 of the vertices of an arbitrary graph G such that $\chi(G-V_1)= 2$ is NP-hard (see [7], page 195). This immediately implies the desired result and completes the proof of the theorem.

CONCLUSIONS

The problems discussed in this paper are motivated by query processing in a distributed computing system. However, the underlying concepts encompass many other applications such as circuit board design [13], distributed system design [14], paging of programs, scheduling, and so on. In graph theoretic terms, these problems are variants of graph partitioning problems.

The recognition that Ghosh's problem of parallel searching of files in a distributed system is a graph theoretic problem enabled us to show that (1) the original problem and some of its extensions are NP-hard, (2) the known heuristic techniques could be employed in solving these problems, and (3) the previously proposed algorithms [9] are of limited utility.

REFERENCES

[1] Appel, K. and Haken, W. "Every Planar Map is Four Colorable," Bulletin American Mathematical Society, Vol. 82, 1976.

[2] Brelaz, D., "New Methods to Color the Vertices of a Graph", Communications of the ACM 22(4), April, 1979, pp. 251-256.

[3] Brooks, R. L. "On Colouring the Nodes of a Network," Proceedings Cambridge Philosophical Society, Vol. 37, 1941, pp. 194-197.

[4] Canaday, R. H., Harrison, R. L., Ivie, E. L., Ryder, J. L., and Wehr, L. A. "A Back-end Computer for Data Base Management," Communications of the ACM, 17 (10), October 1974, pp. 575-582.

[5] Christofides, N. Graph Theory--An Algorithmic Approach, Academic Press, 1975, pp. 58-78.

[6] Eswaran, K. P. "Placement of Records in a File and File Allocation in a Computer Network," Information Processing 74, North-Holland Publishing Co., 1974, pp. 304-307.

[7] Garey, M.R. and Johnson, D.S. Computers and Intractability, W. H. Freeman & Co., 1979.

[8] Garey, M. R., Johnson, D. S., and Stockmeyer, L. "Some Simplified NP-Complete Graph Problems," Theoretical Computer Science, 1, 1976, pp. 237-267.

[9] Ghosh, S. P. "Distributing a Data Base with Logical Associations on a Computer Network for Parallel Searching," IEEE Transactions on Software Engineering, Vol. SE-2, No. 2., June 1976, pp. 106-113.

[10] Hopcraft, J. and Tarjan, R. "Efficient Planarity Testing," Journal of the ACM, 21(4), 1974, pp. 549-568.

[11] Horowitz, E. and Sahni, S. Fundamentals of Computer Algorithms, Computer Science Press Inc., 1978, pp. 501-547.

[12] Karp, R. M. "Reducibility Among Combinatorial Problems," in R. E. Miller and J. W. Thatcher (eds.) Complexity of Computer Computations, Plenum Press, 1972, pp. 85-103.

[13] Kernighan, B. W. and Lin, S. "An Efficient Heuristic Procedure for Paritioning Graphs," The Bell System Technical Journal, Feb. 1970, pp. 291-307.

[14] Lawson, J. T. and Mariani, M. P. "Distributed Data Processing System Design--A Look at the Partitioning Problem," COMPSAC Proceedings, 1978, pp. 358-363.

[15] Matula, D. W., Marble, G., and Isaacson, J. D. "Graph Coloring Algorithms" in Graph Theory and Computing, R. C. Read, (ed), Academic Press, 1972, pp. 109-122.

[16] Parhami, B. "A Highly Parallel Computing System for Information Retrieval," NCC, 1972, pp. 681-690.

[17] Welsh, D.J.A. and Powell, M.B. "An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems," Computer Journal, 10, 1967, pp. 85-86.

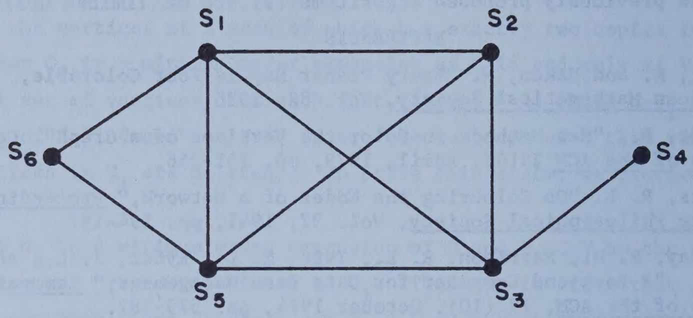

S1
S2
S6
S4
S5
S3

Figure 1. An Example of a Query Graph

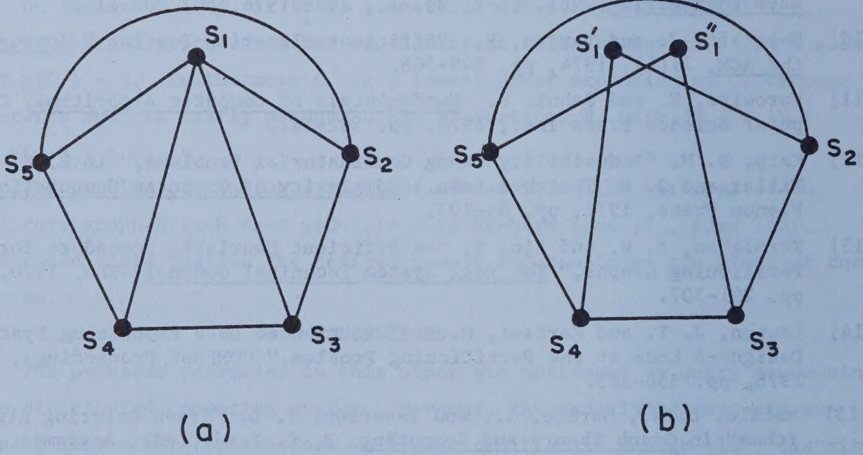

S1
S5
S2
S4
S3
(a)
S1'
S1''
S5
S2
S4
S3
(b)

Figure 2. A Query Graph and its Expansion

MUTUAL EXCLUSION USING INDIVISIBLE READS AND WRITES*

JAMES E. BURNS
Computer Science Department
Indiana University
Bloomington, Indiana 47405

NANCY A. LYNCH
Information & Computer Science
Georgia Inst. of Technology
Atlanta, Georgia 30332

ABSTRACT

The shared space (the number of distinct values taken on by the set of shared variables) requirements of Dijkstra's original mutual exclusion problem are examined. It is shown that 2^N shared states are necessary and sufficient to solve the problem of deadlock-free mutual exclusion for N processes using only individual reads and writes of shared variables for communication.

INTRODUCTION

The first solution to the mutual exclusion problem was given in 1965 by Dijkstra [1]. The original definition of the problem requires that N processes be synchronized so that no two processes are simultaneously executing portions of their code which are called "critical sections". Processes execute asynchronously; that is, they execute at independent, finite, non-zero rates, possibly varying over time. A process may halt if it is not executing its critcal section or the part of its code which is devoted to synchronization. To prohibit trivial solutions, the system of processes is required to be deadlock-free.

In Dijkstra's paper, and subsequent papers by Knuth [2], de Bruijn [3] and Eisenberg and McGuire [4], all communication between processes was required to be through shared variables. The only actions allowed on shared variables were reads and writes, which were assumed to be indivisible.

More recent work has examined the mutual exclusion problem using more powerful primitive actions for accessing shared variables. One of the primary concerns in the papers of Cremers and Hibbard [5], Burns, Fischer, Lynch, Jackson and Peterson [6], Burns [7], Peterson [8], and others has been the number of shared states required to solve certain variations of the mutual exclusion problem. (The number of shared states is the number of distinct values taken on jointly by the set of shared variables.) In this

---

*This research was supported in part by Army Research Office Contract DAAG29-79-C-0155 and NSF Grants MCS77-15628 and MCS78-01689.

paper we examine the number of shared states required for Dijkstra's original problem.

It is interesting to note that although the algorithms in papers [1-4] solve slightly different problems, they all use exactly the same number of shared states, $N*3^N$ (one N-valued variable and N three-valued variables). (Note: Dijkstra [1] uses N pairs of binary variables, but each pair takes on only three possible values.) We will show that N binary variables are necessary and sufficient to solve the problem of deadlock-free mutual exclusion.

READ/WRITE-SYSTEMS

In order to define precisely what we mean by asynchronous systems communicating by shared variables, we will briefly introduce a formal model based on the models presented in [6] and [9]. We have omitted definitions which are not important for the present paper. In the sequel, let $[n] = \{1,2,\ldots,n\}$, for any positive integer n.

A system is a 4-tuple, $S=(P,V,q_0,\phi)$, where P is a finite set of processes, V is a finite set of variables, q_0 is the initial instantaneous description of S and ϕ is the transition function of S. We will always let $N=|P|$ and $M=|V|$ and index the processes of S by [N] and the variables by [M]. The possibly infinite set, X_i, of states of P_i is partitioned into mutually disjoint sets R_i, T_i, C_i and E_i. R_i, T_i, C_i and E_i are called the remainder region, trying region, critical region and exit region, respectively. The set of values of the jth variable of V is V_j. An instantaneous description (id) of S is an $(N+M)$-tuple, $q = (x_1,x_2,\ldots,x_N, v_1,v_2,\ldots,v_M)$, where $x_i \in X_i$ for each $i \in [N]$ and $v_j \in V_j$ for each $j \in [M]$; we use X_i, V_j, and V (ambiguously) as projection operators defined by $X_i(q) = x_i$, $V_j(q) = v_j$ and $V(q) = (v_1,\ldots,v_M)$.

Let Q be the set of all ids of S. Then ϕ is a total function, $\phi: |P| x Q \to Q$, satisfying the following conditions. Let i and j be integers such that $i \neq j$ and $i,j \in [N]$, and let q be an id of S. If $q' = \phi(i,q)$ then we write $q \to q'$ and require that

(1) $X_j(q') = X_j(q)$

(2) If $X_i(q) \in R_i \cup T_i$, then $X_i(q') \in T_i \cup C_i$

(3) If $X_i(q) \in C_i \cup E_i$, then $X_i(q') \in E_i \cup R_i$.

Condition (1) requires that processes communicate only by the shared variables. Conditions (2) and (3) allow processes to loop only within the trying and exit regions, which are used to execute synchronization protocols for the critical region. The internal structure of the remainder and critical regions are not important here and are suppressed.

A system $S=(P,V,q_0,\emptyset)$ is a read/write-system if for every i $\in [N]$, X_i can be partitioned into two disjoint sets Read_i and Write_i such that the following conditions hold for every id q of S. If $X_i(q) \in \text{Read}_i$, then there exists a $k \in [M]$ such that

(4) $V(\emptyset(i,q)) = V(q)$,

(5) for every id q' of S with $X_i(q')=X_i(q)$ and $V_k(q')=V_k(q)$, $X_i(\emptyset(i,q')) = X_i(\emptyset(i,q))$;

we say that the transition $q \rightarrow \emptyset(i,q)$ is a read of the kth variable by P_i. If $X_i(q) \in \text{Write}_i$, then there exists $k \in [M]$ such that

(6) for every $j \in [M]$, if $j \neq k$ then $V_j(\emptyset(i,q)) = V_j(q)$.

(7) for every id q' of S with $X_i(q')=X_i(q)$, $V_k(\emptyset(i,q'))=V_k(\emptyset(i,q))$,

(8) for every id q' of S with $X_i(q')=X_i(q)$, $X_i(\emptyset(i,q')) = X_i(\emptyset(i,q))$;

we say that the transition from q to $\emptyset(i,q)$ is a write of the kth variable by process i and that P_i is about to write the kth variable at q.

In order to reason about the behavior of a system, we wish to specify individual computations, removing the nondeterminism of the transition function of S. This is done with schedules. A schedule of S is a finite or infinite sequence of process indices (elements of [N]). If q_1 is an id of S and $h=i_1i_2\ldots$ $(h=i_1i_2\ldots i_k)$ is an infinite (finite) schedule of S, then $q_1q_2\ldots$ $(q_1\ldots q_{k+1})$ is the computation sequence from q_1 by h, where $q_{j+1} = \emptyset(i_j,q_j)$ for $j = 1,2,\ldots$ $(j=1,\ldots,k)$. If h is finite, then $r(q,h)$ is the final id in the computation sequence from q by h. We say that id q' is reachable from id q if there exists a finite schedule h such that $q' = r(q,h)$. A schedule, h, of S is *admissible from id q* if for every finite prefix, h_1, of h ($h = h_1h_2$), and for every $i \in [N]$, $X_i(r(q,h_1)) \notin R_i$ implies that i occurs in h_2. Admissibility requires that no process halts outside its remainder region.

DEADLOCK-FREE MUTUAL EXCLUSION

A system S satisfies *mutual exclusion* if for every id q of S reachable from q_0 and every pair $i,j \in [N]$, $X_i(r(q,h)) \in C_i$ and $X_j(r(q,h)) \in C_j$ implies that $i=j$.

Process i is said to *change regions* from q by h if there exists finite prefixes h_1 and h_2 of h such that $X_i(r(q,h_1))$ is in a different region than $X_i(r(q,h_2))$. A system S is *deadlock-free* if for every id, q, reachable from q_0 and every non-null schedule, h, admissible from q, some process changes regions from q by h.

THE UPPER BOUND

The algorithm below is in a Pascal-like notation. The algorithm begins execution with all shared variables set to "down" and with process i executing the following program for each $i \in [N]$.

```
program Process i;
   type flag = (down,up);
   shared var F : array [1..N] of flag;
   var j : 1..N;
begin
   while true do begin
   1: F[i] := down;
   2: remainder;       (* remainder region *)
   3: F[i] := down;
   4: for j := 1 to i-1 do
         if F[j] = up then goto 3;
   5: F[i] := up;
   6: for j := 1 to i-1 do
         if F[j] = up then goto 3;
   7: for j := i+1 to N do
         if F[j] = up then goto 7;
   8: critical;        (* critical region *)
      end
   end.
```

Theorem 1: For every integer N>0 there exists a read/write-system of N processes and N binary shared variables which solves the problem of deadlock-free mutual exclusion.

Proof: For any integer $N>0$, it is clear that the above algorithm defines a read/write-system with N processes and N binary shared variables. We must show that the system is deadlock-free and satsifies mutual exclusion.

Suppose deadlock can occur. Then we can reach an id, q, at which at least one process is not in remainder and a schedule h, admissible from q, such that no process changes region from q by h. Since the only backward branches in each process's program occur in the trying region, we observe that for each $i \in [N]$, either $X_i(q) \in R_i$ and i does not occur in h or $X_i(q) \in T_i$ and i occurs infinitely often in h. We call the set of processes which are not in remainder at q "active".

For each $i \in [N]$, define the following subsets of T_i. A_i = the sets of states of P_i corresponding to the statements labeled 3 and 4. B_i = the sets of states of P_i corresponding to the statement labeled 7. We note that if P_i reaches B_i, then it will remain there for the rest of the computation and F[i] will be continuously equal to "up". Let $m = \min \{i \in [N] : P_i$ is active at $q\}$. Since m will eventually detect that no F[i] = up for $i \in [m-1]$, P_m will reach B_m after a finite prefix, h_1, of h ($h = h_1h_2$). (That is, $X_m(q') \in B_m$, where $q' = r(q,h_1)$). After some finite prefix, h_3, of h_2 ($h_2 = h_3h_4$), every active P_i will either be in B_i or will begin cycling forever in A_i with F[i] = down, since all higher indexed processes which do not reach B_i will detect F[m]=up. Let $n = \max \{i \in [N] : X_i(q") \in B_i\}$, where $q"=r(q',h_3)$. Now P_n will find all F[i]=down for $i \in \{n+1,\ldots,N\}$, so P_n will change regions from q" by h_4, contradicting our supposition. Therefore, deadlock cannot occur.

Suppose that mutual exclusion may be violated. Then there must be values $i,j \in [N]$ such that $i \neq j$ and a finite schedule h such that $q=r(q_0,h)$ and $X_i(q) \in C_i$ and $X_j(q) \in C_j$. Let D_i = the set of states of P_i corresponding to statements 6, 7 and 8 of the algorithm, and D_j be similarly defined for P_j. P_i may enter and leave D_i several times before reaching its critical region, but there must be an id at which P_i enters D_i for the last time before going critical. Let q_a be this id for P_i, and let q_b be a similar id for P_j, in the compuation sequence

$q_0 q_1 \ldots q_k$ from q_0 by h ($q=q_k$). We may assume without loss of generality that $a<b$. But then for every c, $a \leq c \leq k$, F[i]=up at q_c. Since P_j must test F[i] after entering D_j (either at statement 6 of 7), P_j cannot go critical in the compuation sequence $q_b q_{b+1} \ldots q_k$, contradicting our supposition. Therefore the algorithm also satisfies mutual exclusion and the theorem is proved. □.

THE LOWER BOUND

Some additional definitions will be needed in the lemmas leading up to the lower bound theorem. As before, we let S be a read/write-system. Let q_1 be an id of S, h be a schedule of S, $i \in [N]$, $v \in V$ and $q_1 q_2 \ldots$ be the computation sequence from q_1 by h. If there exist positive integers $j<k$ such that $q_j \rightarrow q_{j+1}$ is a write of v by P_i and $q_k \rightarrow q_{k+1}$ is a write of v, and if for all n, $j<n<k$, $q_n \rightarrow q_{n+1}$ is not a read of v by any process other than P_i, then we say that the write of v by P_i at q_j is obliterated from q by h.

Let q be an id of S, h be a schedule of S, and $i \in N$. If $X_i(q) \in R_i$ and every write by P_i is obliterated from q by h, then P_i is invisible from q by h. If $i \in [N]$, q is an id of S, h_1, h_2 are schedules of S (h_1 finite) such that P_i is invisible from $r(q,h_1)$ by h_2, then P_i is hidden from q by $h_1 h_2$.

Lemma 1: Let S be a read/write-system, h be a finite schedule of S and P_i be a process of S which is hidden from q_0 by h, and let $q=r(q_0,h)$. Then there is an id q' of S reachable from q_0 such that $X_i(q') \in R_i$, $V(q')=V(q)$ and $X_j(q')=X_j(q)$ for all $j \neq i$.

Proof: Let h_1 be the longest prefix of h for which $X_i(r(q_0,h_1)) \in R_i$ (h_1 exists since P_i is hidden from q_0 by h) so that $h = h_1 h_2$. Let h_3 be the schedule equivalent to h_2 with all occurences of i removed, and let $h' = h_1 h_3$. Now $q'=r(q_0,h')$ meets the requirements of the lemma since P_i cannot have left the remainder region since $r(q_0,h_1)$. Note that $V(q')=V(q)$ because all writes by P_i in the compuation from $r(q_0,h_1)$ to q are obliterated. Also, $X_j(q')=X_j(q)$ for $j \neq i$

because no process can have read anything written by P_i since $r(q_0,h_1)$. □

Let S be a read/write-system, q be an id of S and $i \in [N]$. If P_i is about to write $v \in [M]$ at q, then we say that v is *covered* at q by P_i.

Lemma 2: Let S be a read/write-system with at least two processes which solves deadlock-free mutual exclusion, h be a finite schedule of S and P_i be a process of S hidden from q_0 by h. If P_i goes critical on its own from $q=r(q_0,h)$ by a schedule $h_1 = i^k$, then P_i must write some variable in the computation sequence from q by h_1 which is not covered by any other process at q.

Proof: Suppose P_i goes critical from q by schedule h_1 without writing any variable which is not covered by some other process at q. Let h_2 a schedule consisting of exactly one step of each process other than P_i. Then every write of P_i from q is obliterated from q by h_1h_2, so P_i is hidden from q by h_1h_2. By Lemma 1, there is a reachable id q" which looks like $q'=r(q,h_1h_2)$ to all the other processes but has P_i in remainder. By no deadlock, some other process $P_j \neq P_i$ can go critical from q" by schedule h'. But then $r(q',h')$ has both P_i and P_j critical, contradicting mutual exclusion. □

Let S be a read/write-system, q be an id of S, h be a finite schedule of S and W be a subset of V. We say that W is *nullified from q by h* if for every $w \in W$ there is a process which is hidden from q by h and which covers w at $r(q,h)$.

Lemma 3: Let S be a read/write-system with $N \geq 2$ processes which solves deadlock-free mutual exclusion, and let q be any reachable id of S at which all processes of S are in their remainder regions. For every K, $1 \leq K \leq N$, there is a finite schedule h of S using only processes $P_1, P_2, \ldots, P_K$ such that K variables are nullified from q by h.

Proof: The proof is by induction on K, the number of variables nullified.

BASIS. Let K=1. By no deadlock, there must be a finite schedule h' consisting only of 1's such that P_1 goes critical at r(q,h'). By Lemma 2, there must be a prefix, h" of h' such that P_1 is hidden from q by h" and covers some variable, w, at r(q,h"). But then {w} is nullified from q by h" and the lemma holds for K=1.

INDUCTIVE STEP. Assume Lemma 3 holds for K = k-1. By the inductive assumption, there is a finite schedule h_0 from q using only processes $P_1,\ldots,P_{k-1}$ such that a set, W_1, of k-1 variables is nullified from q_0 by h_0. Let $q_1 = r(q_0,h_0)$. From q_1 we can successively reach id's $q_2,q_3,\ldots$ by finite schedules $h_1,h_2,\ldots$ such that $q_{i+1} = r(q_i,h_i)$, where h_i is defined in the following way. For each i=1,2,..., let h_i begin with the prefix 123...(k-1). From $r(q_i,123\ldots(k-1))$, find an extension of h_i which returns $P_1,\ldots,P_{k-1}$ to their remainder regions (by no deadlock this extension exists). Finally, complete h_i by appending a schedule which nullifies a set, W_{i+1}, of k-1 variables at q_{i+1}. The final portion of h_i exists by the inductive assumption.

For each i=1,2,..., Lemma 1 and Lemma 2 imply that P_k can be moved on its own by a schedule s_i from each q_i such that P_k is ready to write some variable, w_i, which is not in W_i. We claim there are integers i and j, 0<i<j, such that $w_i \notin W_j$. Suppose not. Then $\{w_1,\ldots,w_{i-1}\}$ is a subset of W_i, for i=1,2,... This implies that $w_1,\ldots,w_i$ are distinct since w_i is not in W_i. But then W_{N+2} would contain a subset of N+1 variables, which is impossible (one process is required to nullify each variable) so the claim holds.

Consider the computation from q_i produced by schedule $s = s_ih_ih_{i+1}\ldots h_j$, where i and j are chosen as in the claim. The first part of h_i obliterates all writes of P_k in the computation from q_i by s_i, so P_k is hidden from q_i by s. Now variables in $W_j \cup \{w_i\}$ are nullified from q_i by h_i and $|W_j \cup \{w_i\}| = k$, by the claim. Thus we have k variable nullified from q by $h_0h_1\ldots h_{i-1}s_i\ h_ih_{i+1}\ldots h_j$, and the lemma is proved. ☐

Theorem 2: If S is a read/write-system with at least two processes and S solves deadlock-free mutual exclusion, then S must have at least as many variables as processes.

Proof: By no deadlock, there is an id q reachable from q_0 such that all processes of S are in their remainder regions at q. We may then apply Lemma 3 for q to find another reachable id which covers N distinct variables. □

FURTHER REMARKS

The techniques used in the proof of Lemma 3 may be applied to prove similar theorems for other exclusion problems. The following characteristics appear to be needed. (1) A basis for the induction. It must be possible to show that some number of variables can be covered by a set of hidden processes. (2) For the inductive step, a lemma similar to Lemma 2 must be proved. Secondly, we must have that a subset of N-1 processes of a system of N processes forms a system of the type required. This second condition does not hold for the dining philosophers problem, for example.

Although the algorithm presented in this paper does use fewer shared states than previous solutions, it does not meet one requirement that was imposed by Dijkstra. In [1], Dijkstra states that "The solution must be symmetrical between the N computers; as a result we are not allowed to introduce a static priority". The algorithm given here does not meet this condition because process 1 apparently has the highest priority (it will never be locked out). A future paper will explore the relationships between symmetry and solutions to synchronization problems. One difficulty is finding a useful formal definition of symmetry which coincides with out intuitive understanding of the concept.

Acknowledgement:

We would like to thank Mike Fischer for helpful comment and criticism.

REFERENCES

[1] Dijkstra, E.W. Solution of a problem in concurrent programming control. Comm. ACM 8, 9 (Sep. 1965), 569.

[2] Knuth, D.E. Additional comments on a problem in concurrent control. Comm. ACM 9, 5 (May 1966), 321-322.

[3] de Bruijn, N.G. Additional comments on a problem in concurrent programming control. Comm. ACM 10, 3 (Mar. 1967), 137.

[4] Eisenberg, M.A., and McGuire, M.R. Further comments on Dijkstra's concurrent programming control problem. Comm. ACM 15, 11 (Nov. 1972), 999.

[5] Cremers, A., and Hibbard, T.N. Mutual exclusion of N processors using an O(N)-valued message variable. (extended abstract) Lecture Notes in Computer Science 62, Springer-Verlag (Jul. 1978), 165-176.

[6] Burns, J.E., Fischer, M.J., Jackson, P., Lynch, N.A., and Peterson, G.L. Shared data requirements for implementation of mutual exclusion using a test-and-set primitive. Proc. 1978 Int'l. Conf. on Parallel Processing, Aug. 1978, pp. 79-87.

[7] Burns, J.E. Mutual exclusion with linear waiting using binary shared variables. SIGACT News 10, 2 (Summer 1978), 42-47.

[8] Peterson, G.L. The complexity of parallel algorithms. Ph.D. thesis, Univ. of Washington, 1979.

[9] Burns, J.E. Complexity of communication among asynchronous parallel processes. Ph.D. Thesis, Georgia Inst. of Technology, (in preparation).

IMPROVING ACCESS INTERFERENCE IN DISTRIBUTED DATA BASES

GIANCARLO MARTELLA - FABIO SCHREIBER
Istituto di Elettrotecnica ed Elettronica
Politecnico di Milano
Milano, Italy

Abstract. The storage organization of a Distributed Data Base is examined with particular attention to storage access interference. An "application oriented" organization, particularly suited to certain specialized information systems, is shown to be convenient in terms of network traffic, interference, and response time. Some computational results and properties are given.

1 - INTRODUCTION

The emerging technology of Distributed Databases (DDB) and of Distributed Database Management Systems (DDBMS) has stimulated lots of studies in very different research areas such as Data Models, System Architectures, System Reliability, Data Consistency, Data Integrity, Data Allocation, etc. /BERN 79/ /MENA 79/ /SCHR 79/.

Many of these studies have been carried on in the frame of the design of general purpose DDBMs or with the aim of developing sound theoretical bases for the understanding of the different design choices. However very little has been done from the point of view of the actual requirements which an application, or a class of applications, puts on the functionality of a DDBMS.

Such a study seems rather interesting also considering the performance issues of such systems. Since the growth in complexity due to the distributed environment tends to lower system performance, it should be useful to design different DDBMSs dedicated to different classes of applications with common functional requirements, rather than to put all the most sophisticated techniques into the same general purpose DDBMS.

A first attempt of such a classification of application requirements has been presented in /PAOL 77/ /BALD 79/. In /CERI 80/ the file allocation problem has been solved for a specific environment - constituted by a dedicated network of minicomputers - and optimization has been done with respect to a set of predefined applicative transactions.

In a multi user-multiprogrammed distributed system, one of the crucial points in performance enhancement lies in the solution of problems related to concurrency in data access and to physical memory access interference. On the other hand it seems that many applications - banking systems, Public Administration, etc. - have a strong predominance of local work over remote access requests.

In this paper we are going to discuss memory allocation and access scheduling policies for DDB systems with strong locality properties in order to reduce DB access interference and to enhance response time.

2 - MODELLING THE DDB SYSTEM

Let us consider the DDB system constituted by a set of computer systems — Hosts - each of which is provided with its local set of files (Database) interconnected by a telecommunication network (Fig. 1).

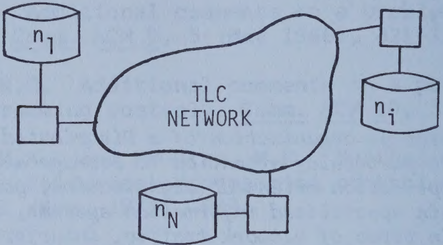


Figure 1

Let it be:

$1 \leq n_i \leq N$	the Hosts of the DDB
$1 \leq m_j \leq M$	the files of the DDB
S_j	the cardinality of file m_j
$1 \leq t_k \leq T$	the transactions from application programs to the DDB
f_{ki}	the frequency of transaction k on host i
ν^r_{kj}	the number of retrieve accesses of transaction t_k on file m_j
ν^u_{kj}	the number of update accesses of transaction t_k on file m_j
C_{loc}	unit cost of a local access
C_{rem}	unit cost of a remote access.

With such a set of parameters, the problem has been solved in /CERI 80/ of improving the locality of reference for a given set of transactions under the constraint of a maximum amount of mass storage at each Host. This solution minimize the following cost function

$$z = LR + RR + LU + RU$$

where

LR = cost of local retrieval = $\sum_{i,j,k} x_{ji} \; f_{ki} \; \nu^r_{kj} \; C_{loc}$

RR = cost of remote retrievals = $\sum_{i,j,k} \overline{x}_{ji} \; f_{ki} \; \nu^r_{kj} \; C_{rem}$

LU = cost of local updates = $\sum_{i,j,k} x_{ji} \; f_{ki} \; \nu^u_{kj} \; C_{loc}$

RU = cost of remote updates = $\sum_{i,j,k} x_{ji} \sum_{\tilde{i} \neq i} f_{k\tilde{i}} \; \nu^u_{kj} \; C_{rem}$

and $x_{ji} = \begin{cases} 0 \text{ if file } m_j \text{ is not allocated on host } n_i \\ 1 \text{ if file } m_j \text{ is allocated on host } n_i. \end{cases}$

In the solution, the isomorphism with the well-known Knapsack problem has been used.

Let us call simple transaction a transaction which needs a single access to a data file (i.e. $\nu_k^r + \nu_k^u = 1$) and let us now consider the probability p_{ij} that a simple transaction t_i, processed at host n_i, accesses the file m_j at host j. If we look at each file as the server of a queue, we can characterize the whole system as a discrete-time finite Markov chain the states of which are defined as a set of 2N-tuples $(m_1, k_1, \ldots, m_N, k_N)$, where k_i is the position in the queue of file m_i of the request from transaction t_i. Such a model has been already discussed by Baskett and Smith /BASK 76/ for memory interference in multiprocessor systems. They demonstrated that if references from processors to memory modules are made randomly with a uniform probability (interleaving)

$$\forall_i, \forall_j \quad p_{ij} = \frac{1}{M}$$

the expected memory utilization is

$$\rho = 1/2\ (((2N-1)/M + 2) - (((2N-1)/M + 2)^2 - 8N/M)^{1/2})$$

which demonstrates a loss from 25% up to 41% of processing power due to memory interference.

This result again is in favor of an allocation of files which is dependent on the type of application work in the sense of maximizing local references with respect to remote accesses.

3 - EVALUATING THE RESPONSE TIME

The similarity with the problem studied in /BASK 76/ is even closer if we consider that in DDB systems complex transactions can be statically decomposed into simple ones; since the allocation of files is also static, the probabilities p_{ij} can be easily evaluated from the knowledge of f_{ki}, ν_{kj}^r and ν_{kj}^u.

Since the allocation of files has been made with the aim of being optimal with respect to a given class of applications /CERI 80/, the probabilities p_{ij} of referencing files which are stored at the same host as the transactions themselves are much higher than those referencing remote files. Therefore the system behaves like a "Home memory system with a static overflow placement" described by Smith /SMIT 77/. In our case, transactions stand in place of processors, simple transactions stand in place of memory references, and interference in memory modules usage is replaced by access to a file in an exclusive mode.

In particular Theorems 1 and 2 of /SMIT 77/ hold, which say respectively that memory interference at each site is minimized if the queuing policy of the local DBMS is such as to give priority to access requests coming from remote transactions and that, among these requests, priority must be given to those having the lowest access rate.

We can choose then a HOL (Head of the Line) /KLEI 76/ priority rule as the queuing discipline for each local DBMS. The mean servicetime $\overline{x}$ is the same for all the customers and corresponds to the time required for one logical access request to the DB. (Fig. 2)

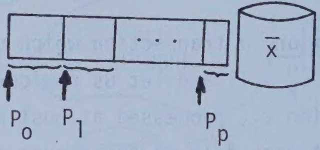


Figure 2

The lowest priority level P is given to transactions originated at the same node the file is stored at; higher priority levels (P_1 ... P_p) are given to transactions coming from outside in an order which is inverse to their access rate to the file, that is:

$$\forall\ t_i,\ t_j\ \ \exists\ \lambda_i,\ \lambda_j \rightarrow \lambda_i < \lambda_j \Leftrightarrow P_{(t_i)} > P_{(t_j)}$$

if the case happens that

$$\lambda_i = \lambda_j$$

then the priority rule is the FCFS (First Come First Served).

The mean arrival rates at a server file m_j for each simple transaction are evaluated as follows

$$\lambda_k = f_{ki} * (\nu^r_{kj} + \nu^u_{kj})$$

Then we can evaluate the average response time $T_p = W_p + \overline{x}$ for a transaction in priority class p, where W_p is the average waiting time for class p and it is given (see /KLEI 76/) by:

$$W_p = W_o + \overset{P}{\underset{p}{\Sigma_i}}\ \overline{x}\ \lambda_i\ W_i + \overset{P}{\underset{p+1}{\Sigma_i}}\ \overline{x}\ \lambda_i\ W_p = \frac{W_o + \overset{P}{\underset{p+1}{\Sigma_i}}\ \rho_i\ W_i}{1 - \overset{P}{\underset{p}{\Sigma_i}}\ \rho_i}$$

where $W_o = \overset{P}{\underset{1}{\Sigma_i}} \dfrac{\lambda_i\ \overline{x}^2}{2}$ is the delay due to the transaction being served when the considered transaction enters the queue, and ρ_i is the fraction of time the server is busy with elements of class i. A recursive solution method can be applied /KLEI 76/ and we obtain

$$W_p = \frac{W_o}{(1-\sigma_p)\,(1-\sigma_{p+1})} \qquad \text{where } \sigma_p = \sum_{i=p}^{P} \rho_i \qquad p = 1,2,\ldots,P$$

4 - EVALUATING THE WASTED CYCLES

Let us suppose now that accessing a file on the same Host i takes an average of q_i unit cycles, where a unit cycle is the local access time, which is supposed to be independent from the file structure, while accessing a remote file j takes q_j cycles.Then the remote access "wastes" q_j-1 $(j\neq i)$ cycles of the requesting Host and 1 cycle of the servicing Host, subtracting q_j cycles from the productivity of the whole system.

q_j is proportional to the number of waiting requests at node j which preceed the request from i, which has a priority level P_i. Then /KLEI 76/

$$q_j = \sum \overline{N}_{kp} + \overline{M}_{kp} = \{ \sum_k \lambda_k W_k + \sum_\ell \lambda_\ell W_p \} \qquad \left.\begin{array}{l} k = p,p+1,\ldots,P \\ \ell = p+1,\ p+2,\ldots,P \end{array}\right\} \text{ on node } j$$

Let us define the rate of a transaction i as

$$\text{Rate}_i = [p_{ii} \cdot q_i + \sum_{j\neq i} p_{ij} \cdot q_j]^{-1}$$

Then the total number of wasted cycles per unit time due to the interference among transactions is:

$$\text{Total Wasted Cycles} = \sum_i \text{Rate}_i \cdot \sum_{j\neq i} p_{ij}\, q_j$$

5 - COMPUTING RESULTS

Figures 3 to 6 show some computing results for a Distributed DB with 6 local nodes. In particular in Fig. 3 the normalized response time is plotted for each priority class as a function of the percentage of local and remote work. The 0 priority is given to local work while the other 5 classes are distributed among remote work. The remote accesses of a transaction are uniformly distributed among the nodes ($\sigma = 0$). Fig. 4 shows the same function but in the case of a random distribution of remote accesses such as to privilege a node with respect to the others($\sigma = 39.7$).

Fig. 5 shows respectively the normalized maximum response time for three different scheduling policies.in the case of $\sigma= 0$. The first gives the priority only on the base of the access rate (λ-dependent) independently of the access being local or remote. The second gives the lowest priority to the local access requests, while the third gives the highest priority to local accesses. Fig. 6 shows the same results for $\sigma= 39.7$.

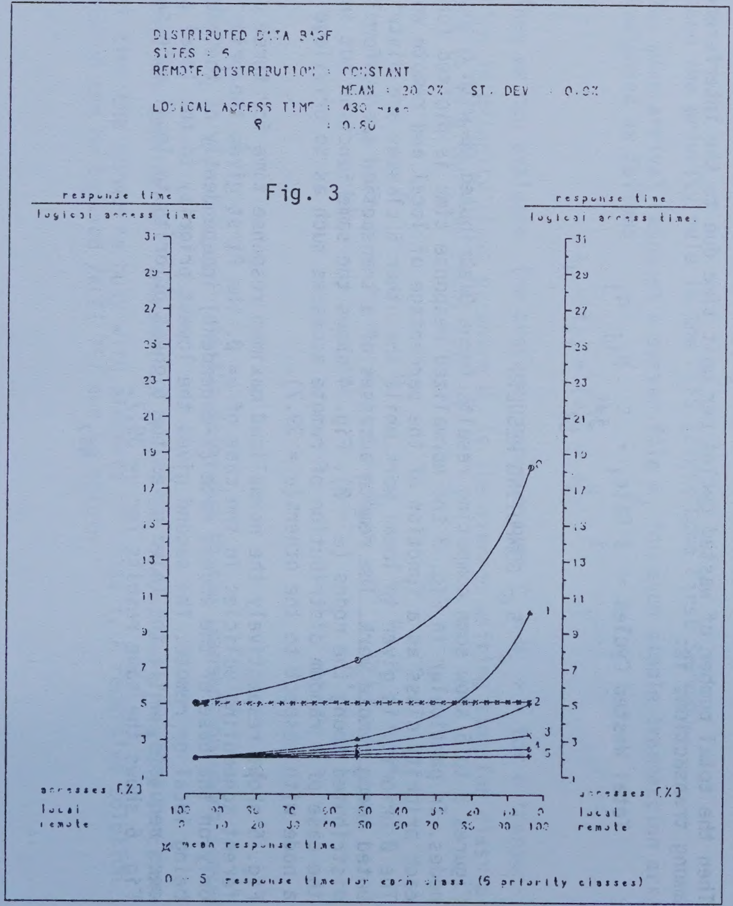


Fig. 3

DISTRIBUTED DATA BASE
SITES : 6
REMOTE DISTRIBUTION : CONSTANT
MEAN : 20.0% ST. DEV : 0.0%
LOGICAL ACCESS TIME : 430 msec
ρ : 0.60
response time / logical access time
31
29
27
25
23
21
19
17
15
13
11
9
7
5
3
1
0
1
2
3
4
5
accesses [%]
local
remote
100 90 80 70 60 50 40 30 20 10 0
0 10 20 30 40 50 60 70 80 90 100
x mean response time
0 - 5 response time for each class (6 priority classes)

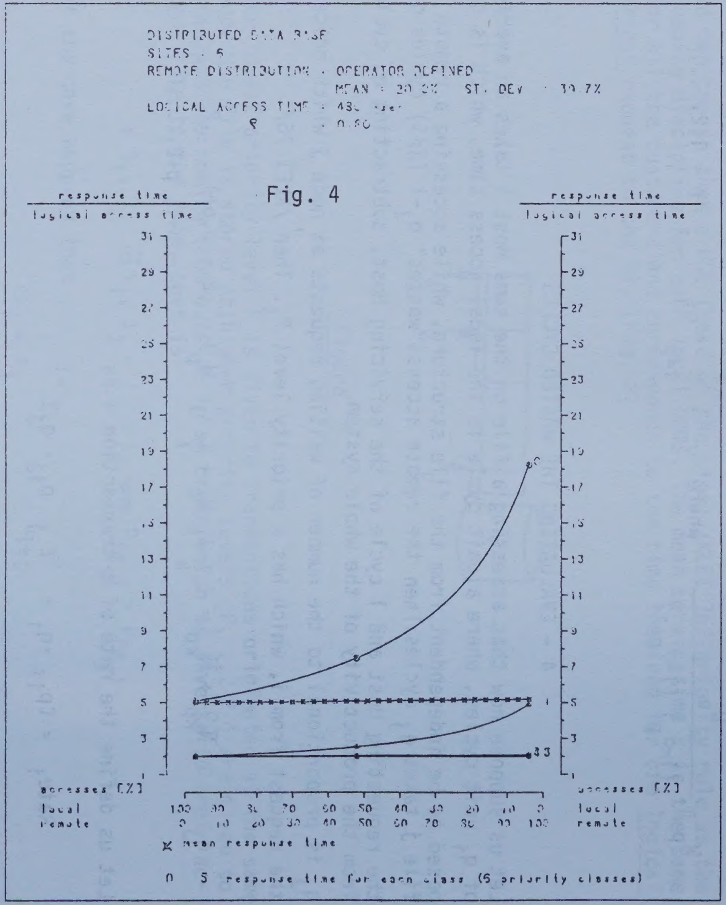


Fig. 4

DISTRIBUTED DATA BASE
SITES : 6
REMOTE DISTRIBUTION : OPERATOR DEFINED
MEAN : 20.0% ST. DEV : 39.7%
LOGICAL ACCESS TIME : 430 msec
ρ : 0.60
response time / logical access time
31
29
27
25
23
21
19
17
15
13
11
9
7
5
3
1
0
1
3 3
accesses [%]
local
remote
100 90 80 70 60 50 40 30 20 10 0
0 10 20 30 40 50 60 70 80 90 100
x mean response time
0 - 5 response time for each class (6 priority classes)

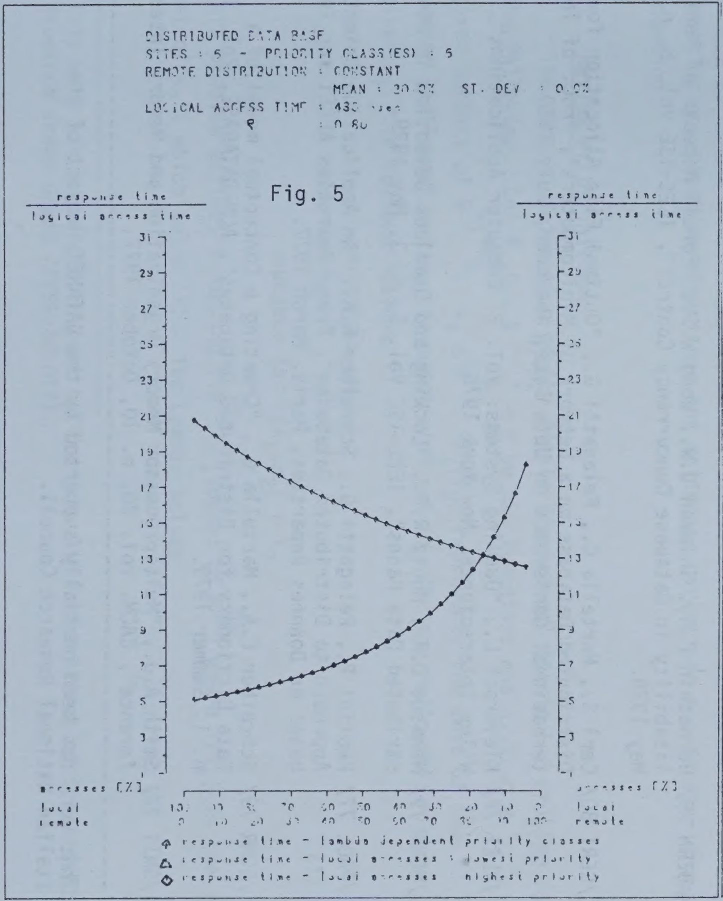


Fig. 5
DISTRIBUTED DATA BASE
SITES : 5 - PRIORITY CLASS(ES) : 6
REMOTE DISTRIBUTION : CONSTANT
MEAN : 20.0% ST. DEV. : 0.0%
LOGICAL ACCESS TIME : 430 msec
ρ : 0.80
response time / logical access time
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
accesses [%]
local 100 90 80 70 60 50 40 30 20 10 0
remote 0 10 20 30 40 50 60 70 80 90 100
response time - lambda dependent priority classes
response time - local accesses : lowest priority
response time - local accesses : highest priority

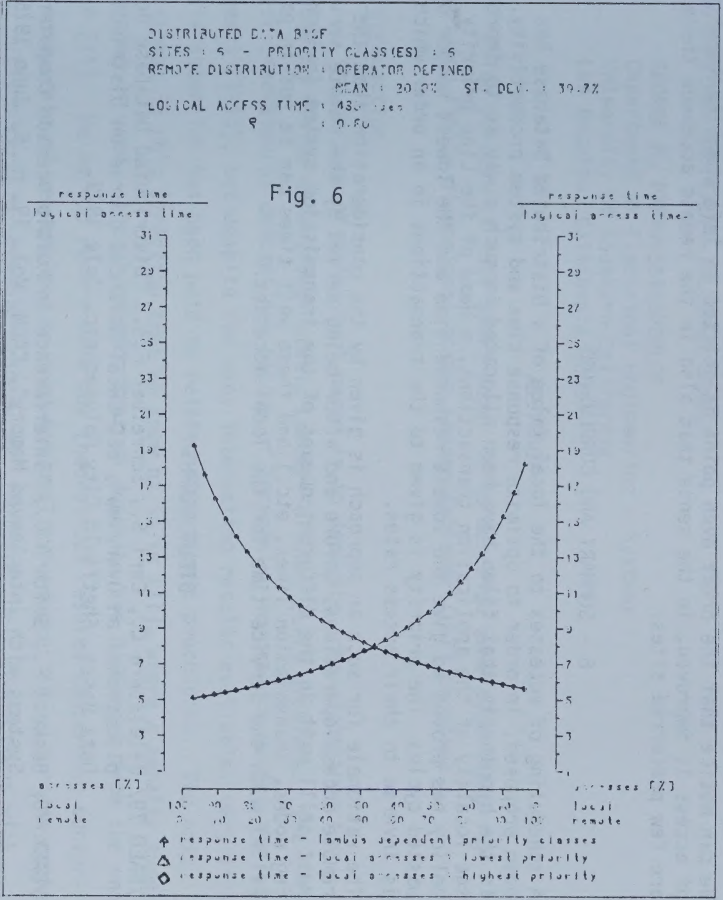


Fig. 6
DISTRIBUTED DATA BASE
SITES : 6 - PRIORITY CLASS(ES) : 6
REMOTE DISTRIBUTION : OPERATOR DEFINED
MEAN : 20.0% ST. DEV. : 39.7%
LOGICAL ACCESS TIME : 430 msec
ρ : 0.60
response time / logical access time
1
3
5
7
9
11
13
15
17
19
21
23
25
27
29
31
accesses [%]
local 100 90 80 70 60 50 40 30 20 10 0
remote 0 10 20 30 40 50 60 70 80 90 100
response time - lambda dependent priority classes
response time - local accesses : lowest priority
response time - local accesses : highest priority

We can notice that the break even point is shifted to left when the locality of access is improved, in the sense that also in the remote accesses there are few preferred sites.

6 - SUMMARY AND CONCLUSIONS

A scheduling of accesses to the local files of a Distributed Database has been proposed, in order to optimize response time and system productivity. In the hypothesis that files have been allocated in such a way as to improve the locality of the application transactions, a Head of the Line priority policy has proved to give the lowest response time and the lowest number of wasted cycles. The priority is given to the transactions in an order which is inverse to their access rates.

The rationale for such an approach is given by the consideration that the remote access operations, before and after being served by the remote server, must still wait in the different queues of the transmission system (switching nodes, transmission lines, etc.) and these wait times can be put in parallel with the service time for the local accesses.

BIBLIOGRAPHY

/BALD 79/ Baldissera C., Ceri S., Schreiber F.A., "Distributed Database Applications: an Overview", EEC-CREST Advanced Course on Distributed Data Bases, Sheffield City Polytechnic, July 1979.

/BASK 76/ Baskett F., Smith A.J., "Interference in Multiprocessor Computer Systems with Interleaved Memory", CACM, Vol. 19, n. 6, June 1976.

/BERN 79/ Bernstein P.A., Shipman D.W., Wong W.S., "Formal Aspects of Serializability in Database Concurrency Control", IEEE-TSE Vol. 5, n. 3, May 1979.

/CERI 80/ Ceri S., Martella G., Pelagatti G., "Optimal File Allocation for a Distributed Database on a Network of Minicomputers", Proc. of International Conference on Data Base, Aberdeen, July 1980.

/KLEI 76/ Kleinrock L., "Queuing Systems: Vol. 2° Computer Applications", Wiley Interscience, New York 1976.

/MENA 79/ Menasce D.A., Muntz R.R., "Locking and Deadlock Detection in Distributed Data Bases", IEEE-TSE Vol. 5, n. 3, May 1979.

/PAOL 77/ Paolini P., Pelagatti G., Schreiber F.A., "An Application Oriented Approach to Distributed Databases", Proc. Journéées AFCET sur les Bases de Donnéés Reparties, Paris, March 1977.

/SCHR 79/ Schreiber F.A., Martella G., "Creating a Conceptual model of a Data Dictionary for Distributed Databases", ACM-DATABASE, Vol. 11, n. 1, Summer 1979.

/SMIT 77/ Smith A.J., "Multiprocessor Memory Organization and Memory Interference", CACM, vol. 20, n. 10, October 1977.

This work has been partially supported by the DATANET Project of the Italian National Research Council.

ALGORITHMS FOR LEADER-FOLLOWER GAMES*

GEORGE P. PAPAVASSILOPOULOS
Department of Electrical Engineering — Systems
University of Southern California
Los Angeles, California 90007

ABSTRACT

The purpose of this short paper is to describe some algorithms for solving finite dimensional static Leader-Follower (L-F) problems.

Introduction

The purpose of this short paper is to describe some algorithms for solving finite dimensional static Leader-Follower (L-F) problems. The L-F (or Stackelberg) concept has become of increasing importance during the last years [1-10], but despite the many interesting results available, very little attention has been paid to related algorithmic procedures. The only attempt in this direction that we know of is in [11], where a penalty type method is used. The algorithms that we describe are different from the one of [11]. We do not report here any theoretical results concerning convergence, rate of convergence or computational experience; it is our aim rather to outline certain sensible procedures, indicate their connections with existing optimization algorithms and delineate directions for further investigation.

1. Review of Definitions

The general definition of the finite dimensional static L-F problem is the following. Let $f_1,f_2: R^{n_1+n_2} \to R$ be two functions and $X_1,X_2 \subseteq R^{n_1+n_2}$ given subsets of $R^{n_1+n_2}$. $x_i \in R^{n_i}$. $f_1(f_2)$ represents the cost of the Leader (Follower) and X_1 his constraint set (X_2 for the Follower). For given x_1, the Follower solves

$$\begin{aligned} &\min_{x_2} f_2(x_1,x_2) \\ &\text{subject to: } (x_1,x_2) \in X_2 \end{aligned} \tag{1}$$

This problem defines a mapping $T:R^{n_1} \to \{\text{subset of } R^{n_2}\}$ where $T(x_1)$ is the set of x_2's which solve (1). The Leader solves

---

*This work was supported in part by the U.S. Air Force Office of Scientific Research under Grant AFOSR-80-0171.

$$\begin{aligned} &\min_{x_1,x_2} f_1(x_1,x_2) \\ &\text{subject to: } (x_1,x_2) \in X_1 \\ &\qquad\qquad\quad x_2 \in T(x_1). \end{aligned} \tag{2}$$

The final aim of the L-F problem is to solve (2). To avoid the situation where $(\bar{x}_1,\bar{x}_2)$ solves (2), and the Follower chooses $\tilde{x}_2 \in T(\bar{x}_1)$ with $(\bar{x}_1,\tilde{x}_2) \notin X_1$, we will assume that $X_2 \subseteq X_1$. X_i can be a subset of $R^{n_1+n_2}$ determined through equality or inequality constraints.

2. Algorithms

(i) First Class: Let $X_1 = X_2$ be determined by $h(x_1,x_2) \le 0$, where h is a function $R^{n_1+n_2} \to R$. Then under (smoothness, regularity or convexity) assumptions, (2) is equivalent to solving

$$\begin{aligned} &\min f_1(x_1,x_2) \\ &\text{subject to: } \frac{\partial f_2(x_1,x_2)}{\partial x_2} + \mu \frac{\partial h(x_1,x_2)}{\partial x_2} = 0 \\ &\qquad\qquad\quad \mu \ge 0 \\ &\qquad\qquad\quad h(x_1,x_2) \le 0. \end{aligned} \tag{3}$$

(3) is a classical nonlinear programming problem with unknowns x_1,x_2,μ. Application of the known methods will (usually) require that f_2 and h are twice continuously differentiable. Solving (3) might not be so appealing, since we have increased the number of unknowns and constraints. This is not so bad since we cannot get rid of the constraint $x_2 \in T(x_1)$ without any punishment. The important issue is to employ a nonlinear programming algorithm for (3) which will take advantage of the underlying L-F problem philosophy. Such algorithms are generated in the second class below.

(ii) Second Class: Let $X_1 = X_2$ be determined by $h(x_1,x_2) = 0$. Since for given x_1, the Follower solves (2), it is natural to generate an algorithm based on the following rational: For $x_1 = x_1^k$, the Follower solves (2) and finds $x_2 = x_2^k$. The Leader, knowing x_1^k, x_2^k, moves to $x_1^{k+1} = x_1^k + d_k$ and the Follower solves (2) with $x_1 = x_1^{k+1}$ and finds $x_2^{k+1} = x_2^k + c_k$. The Leader would like to choose d_k, so that $f_1(x_1^k, x_2^k) > f_1(x_1^{k+1}, x_2^{k+1})$, i.e. to improve his cost in the next iteraction. It is reasonable to assume that for d_k small, so is c_k. (In what follows appropriate regularity or convexity conditions and sufficient smoothness of f_1,f_2,h are tacitly assumed. Also for simplicity reasons let

$n_1 = n_2 = 1$.) Since ${x_2}^k$ solves (2) when $x_1 = {x_1}^k$, it will hold

$$\frac{\partial f_2({x_1}^k,{x_2}^k)}{\partial x_2} + \mu^k \frac{\partial h({x_1}^k,{x_2}^k)}{\partial x_2} = 0 \qquad (4)$$

$$h({x_1}^k,{x_2}^k) = 0$$

Similarly

$$\frac{\partial f_2({x_1}^k+d_k,{x_2}^k+c_k)}{\partial x_2} + \mu^{k+1} \frac{\partial h({x_1}^k+d_k,{x_2}^k+c_k)}{\partial x_2} = 0 \qquad (5)$$

$$h({x_1}^k+d_k,{x_2}^k+c_k) = 0.$$

μ^k,μ^{k+1} are Lagrange multipliers. A first order approximate expansion of (5), around ${x_1}^k,{x_2}^k$ and use of (4) yields

$$0 = \frac{\partial^2 f_2({x_1}^k,{x_2}^k)}{\partial x_1 \partial x_2} d_k + \frac{\partial^2 f_2({x_1}^k,{x_2}^k)}{\partial {x_2}^2} c_k$$

$$+ \mu^{k+1} \left[\frac{\partial^2 h({x_1}^k,{x_2}^k)}{\partial x_1 \partial x_2} d_k + \frac{\partial^2 h({x_1}^k,{x_2}^k)}{\partial {x_2}^2} c_k \right] + (\mu^{k+1}-\mu^k) \frac{\partial h({x_1}^k,{x_2}^k)}{\partial x_2} \qquad (6a)$$

$$\frac{\partial h({x_1}^k,{x_2}^k)}{\partial x_1} d_k + \frac{\partial h({x_1}^k,{x_2}^k)}{\partial x_2} c_k = 0 \qquad (6b)$$

Since we want $f_1({x_1}^{k+1},{x_2}^{k+1}) < f_1({x_1}^k,{x_2}^k)$, expanding $f_1({x_1}^k+d_k,{x_2}^k+c_k)$ around ${x_1}^k$, ${x_2}^k$ we obtain

$$\frac{\partial f_1({x_1}^k,{x_2}^k)}{\partial x_1} d_k + \frac{\partial f_1({x_1}^k,{x_2}^k)}{\partial x_2} c_k \quad 0 \qquad (7)$$

If one has an efficient way of solving the system (6a), (6b), (7) for d_k, c_k, μ^{k+1} then one can determine d_k. The bad thing with this is that (6a) is nonlinear since μ^{k+1} multiplies d_k and c_k. Of course one could choose μ^{k+1} to be equal to μ^k, which μ^k is known presumably from the previous minimization of (2) and try to find d_k,c_k as to make the left hand side of (7) as small as possible. (Conditions like $\|d_k\| \leq 1$ $\|c_k\| \leq 1$ should be imposed in doing that.) If the second order sufficiency conditions for (2) are satisfied at $({x_1}^k,{x_2}^k)$ and μ^{k+1} is close to μ^k, it might hopefully hold

$$\frac{\partial^2 f_2({x_1}^k,{x_2}^k)}{\partial {x_2}^2} + \mu^{k+1} \frac{\partial^2 h({x_1}^k,{x_2}^k)}{\partial {x_2}^2} = A_k > 0 \qquad (8)$$

(since μ^{k+1} will be close to μ^k) and thus solving (6) for c_k we obtain

$$c_k = -A_k^{-1}\left[\frac{\partial^2 f_2(x_1{}^k,x_2{}^k)}{\partial x_1 \partial x_2} + \mu^{k+1}\,\frac{\partial^2 h(x_1{}^k,x_2{}^k)}{\partial x_1 \partial x_2}\right] d_k +$$

$$-A_k^{-1}(\mu^{k+1}-\mu^k)\,\frac{\partial h(x_1{}^k,x_2{}^k)}{\partial x_2} \tag{9}$$

Substituting c_k from (9) into (7) and (6b) yields

$$\left\{\frac{\partial f_1}{\partial x_2} - A_k^{-1}\,\frac{\partial f_1}{\partial x_1}\left[\frac{\partial^2 f_2}{\partial x_1 \partial x_2} + \mu^{k+1}\,\frac{\partial^2 h}{\partial x_1 \partial x_2}\right]\right\} d - \frac{\partial f_1}{\partial x_2}\,A_k^{-1}(\mu^{k+1}-\mu^k)\,\frac{\partial h}{\partial x_2} < 0 \tag{10a}$$

$$\left\{\frac{\partial h}{\partial x_1} - \frac{\partial h}{\partial x_2}\,A_k^{-1}\left[\frac{\partial^2 f_2}{\partial x_1 \partial x_2} + \mu^{k+1}\,\frac{\partial^2 h}{\partial x_1 \partial x_2}\right]\right\} d - \frac{\partial h}{\partial x_2}\,A_k^{-1}(\mu^{k+1}-\mu^k)\,\frac{\partial h}{\mu x_2} = 0 \tag{10b}$$

A full rank assumption on $\frac{\partial h}{\partial x_2}$ in conjunction with the invertibility of A_k makes (10b) solvable in $\mu^{k+1}-\mu^k$, in which case substitution of $\mu^{k+1}-\mu^k$ in (10a) results in a single inequality condition for d_k. Of course A_k depends on μ^{k+1}, but we can set μ^{k+1} equal to μ^k while calculating A_k and use also μ^k in place of μ^{k+1} in the first term of (10a). Thus we end up with a single inequality for d_k through which d_k can be determined, by using $x_1{}^k, x_2{}^k, \mu^k$ only.

A little reflection will persuade the reader that the method described above (after (8)) is a variation of the reduced gradient projection method for the problem (3), where the unknowns are x_1, x_2, μ. The difference between this method and the reduced gradient method as described in [13] is essentially the following. In [13] the problem

$$\begin{aligned} &\min f(y_1,y_2) \\ &\text{subject to: } h(y_1,y_2) = 0 \end{aligned} \tag{11}$$

is considered and $\frac{\partial h}{\partial y_2}$ (mxm matrix) is assumed nonsingular. y_1 is considered as the independent variable and then $y_2 = g(y_1)$ (since $\frac{\partial h}{\partial y_2}$ nonsingular). (g is not found explicitly.) The procedure of [13] solves essentially:

$$\min f(y_1,g(y_2))$$

In the algorithm we described, the situation is as follows. Starting from

$$\begin{aligned} &\min f(y_1,y_2) \\ &\text{subject to: } h_1(y_1,y_2,y_3) = 0 \\ &\qquad\qquad\quad h_2(y_1,y_2) = 0 \end{aligned} \tag{12}$$

((12) corresponds to (3) and (y_1,y_2,y_3) corresponds to (x_1,x_2,μ) we consider $\frac{\partial h_1}{\partial y_2}$ (mxm) nonsingular (see (8)) and thus $y_2 = g(y_1,y_3)$ for some g. So, we solve essentially

$$\begin{aligned} &\min f(y_1,g(y_1,y_3),y_3) \\ &\text{subject to: } h_2(y_1,g(y_1,y_3),y_3) \end{aligned} \tag{13}$$

In other words whereas in (11) ([13]) the constraint is implicitly eliminated, the algorithm we described for the L-F problem eliminates only part of the constraints of (3) (see (12),(13)).

Considering the difficulty of handling gradient projection methods and the intricasies concerning proofs about their convergence, rate of convergence and so on, one might feel not optimistic concerning the algorithm we described. It should not be forgotten though that the partial elimination of constraints method that we essentially described, might very well be suited to several Leader-Follower problems, because perhaps of the type of functions involved and because it springs up from the philosophy underlying L-F games as such.

It should also be noticed that if (7) had been substituted by

$$\underset{c_k,\, d_k}{\text{minimize}}\ \frac{\partial f_1}{\partial x_1} d_k + \frac{\partial f_1}{\partial x_2} c_k + \tfrac{1}{2}\begin{bmatrix} d_k \\ c_k \end{bmatrix}' [\nabla^2_{x_1x_2} f_1] \begin{bmatrix} d_k \\ c_k \end{bmatrix} \tag{14}$$

Subject to: (6a),(6b)

we would have ended up with a Newton-type iteration on the submanifold of the constraints.

A Special Case of the Second Class: Linear Case: In case f_1,f_2 are linear and the constraint set $X_1 = X_2$ is a polyhedron described by linear inequality constraints, the philosophy underlying the algorithm presented yields a particularly attractive procedure. Let $f_1(x_1x_2) = c_1'x_1 + c_2'x_2$, $f_2(x_1,x_2) = d_1'x_1 + d_2'x_2$ and $X_1 = X_2$ be determined by $A_1x_1 + A_2x_2 \le b$. c_i, d_i, b are constant vectors and A_1, A_2 constant matrices. Let us assume that the constraint set is bounded. (2) assumes the form

$$\begin{aligned} &\min c_1'x_1 + c_2'x_2 \\ &\text{subject to: } A_1x_1 + A_2x_2 \le b \end{aligned} \tag{15}$$

$$d_2'x_2 \le d_2'\bar{x}_2,\ \forall \bar{x}_2 \text{ with } A_1x_1 + A_2\bar{x}_2 \le b$$

$T(x_1)$ lies on the closure of the constraint polyhedron and thus the whole reaction set $\mathcal{R}$ of the Follower (i.e., the union of $T(x_1)$'s for all x_1's) lies on the surface of the polyhedron. Thus we can be moving on the outer surface of the polyhedron. Our aim is to find a point in $\mathcal{R}$ which admits a supporting hyperplane perpendicular to (c_1,c_2) leaving $\mathcal{R}$ on the opposite direction of the one where $(-c_1,-c_2)$ points. These geometric considerations suggest the following algorithm.

Step 1: For given x_1 we solve the Followers problem and find a solution x_2.

Step 2: At x_2, we examine all the edges passing from it (finite in number) as to find those having a negative angle with (c_1,c_2). Call $(x_1^1,x_2^1)\ldots(x_1^e,x_2^e)$ the extreme points which are the ends of those directions of decent for f_1, emanating from (x_1,x_2).

Step 3: Find which ones out of $(x_1^1,x_2^1),\ldots,(x_1^e,x_2^e)$ are stationary points for d_2x_2 when we restrict our attention to the hyperplane $x_1 = x_1^1$(or...or x_1^e) (By stationary we mean that it solves min $d_2'x_2$ subject to $A_1x_1 + A_2x_2 \leq b$ where $x_1 = x_1^1$ (or...or x_1^e)). If x_2^m is stationary, let $x_1 = x_1^m$ and go to Step 1. If none is stationary stop.

This algorithm will terminate effectively in a finite number of iterations, but it might converge to a point which is a local minimum of (2), because the reaction set $\mathcal{R}$ is not necessarily convex. The nonconvexity of $\mathcal{R}$ appears although f_1,f_2 and the constraints are linear (convex). The nonconvex character of $\mathcal{R}$ is not easy to deal with. Of course under conditions which guarantee convexity (or even less a kind of monotonicity) of $\mathcal{R}$ this is taken care of. One such case is the one where $d_2 = c_2$, where the Leaders problem can be treated as a team problem; i.e. to solve the L-F game we just minimize the Leaders cost subject to x_1,x_2 in the constraint set and forget about the Followers cost. (This can be done also if $f_1(x_1,x_2) = f_2(x_1,x_2) + g(x_1)$ with f_1,f_2,g some nonlinear functions, i.e., when the Leaders cost is the sum of the Followers cost and of another cost which depends on the Leader's decision only.)

(iii) Third Class: This algorithm is motivated by the algorithms presented in [12]. The algorithm of [12] deals with the problem

$$\begin{aligned} &\min f(x) \\ &\text{subject to: } g(x,y) \le 0 \quad \forall y \in Y \\ &\qquad\qquad\qquad x \in X \end{aligned} \tag{16}$$

where Y is an infinite (compact) subset of $R^n (x \in R^n, y \in R^m)$. The idea in [12] is to choose α finite subset $Y^k \subseteq Y$ and substitute $g(x,y) \le 0 \;\; \forall y \in Y$ with $g(x,y) \le 0 \;\; \forall y \in Y^k$; after solving the new problem which has a finite number of constraints, alter Y^k to Y^{k+1} and solve a new problem with finite number of constraints again. An intelligent construction of the Y^k's leads to the solution of the initial problem by solving a sequence of problems with finite number of constraints. This scheme applies directly to the L-F problem if the constraints for x_1, x_2 are not coupled; i.e., $x_1 \in X_1 \subseteq R^{n_1}$, $x_2 \in X_2 \subseteq R^{n_2}$, where X_2 will assume the role of Y in the algorithm of [12]. Nonetheless, if the L-F problem is subjected to a constraint of the form $h(x_1,x_2) \le 0$, then (2) becomes

$$\begin{aligned} &\min f_1(x_1,x_2) \\ &\text{subject to: } h(x_1,x_2) \le 0 \\ &\qquad f_2(x_1,x_2) \le f_2(x_1,x_3) \quad \forall x_3 \ni: \; h(x_1,x_3) \le 0 \end{aligned} \tag{17}$$

and thus the set Y depends on x_1, i.e., $Y(x_1) = \{x_2 : h(x_1,x_2) \le 0\}$. Thus, one should generalize the scheme of [12] as to handle the problem

$$\begin{aligned} &\min f(x) \\ &\text{subject to: } g(x,y) \le 0 \quad \forall y \in Y(x) \\ &\qquad\qquad\qquad x \in X \end{aligned} \tag{18}$$

if he wishes to employ the rational of [12]. The choice of Y^k is quite difficult since Y^k will depend on x. It is not obvious to us how one can do that. In any case, if the constraints of the L-F problem are of the form $x_1 \in X_1 \subseteq R^{n_1}$, $x_2 \in Y_2 \subseteq R^{n_2}$, then use of [12]'s algorithm is possible. Essentially the same discussion applies to the algorithm suggested in [14] for handling problems with an infinite number of constraints.

Conclusions

The three types of classes of algorithms that we described qualitatively above, seem to provide promising directions along which a detailed construction and study of L-F algorithms might go. A quite difficult task concerning the L-F algorithms is due to the nonconvex character of the Followers reaction set. Of course difficulties due to nonconvexity appear in many optimization problems and one cannot expect the researchers interested in L-F algorithms to resolve the "nonconvex" issues. What is reasonable though is to consider subclasses of L-F problems (such as those where the Leaders problem is essentially a team problem, see end of discussion on the Second Class of Algorithms) and prove global convergence of the algorithms under appropriate (convexity ?) assumptions on the constraints and cost which yield a nicely behaved reaction set. The study of local convergence of the second class algorithms can benefit by knowledge concerning gradient projection methods [13].

As far as it concerns the algorithms of the third class it would be worthy to try to bypass the dependence of Y on x keeping still in force the ideas of [12] or [14].

REFERENCES

[1] J.B. Cruz, Jr., "Leader-Follower Strategies for Multilevel Systems," IEEE Trans. on Automatic Control, Vol. AC-23, No. 1, 1978.

[2] C.I. Chen and J.B. Cruz, Jr., "Stackelberg Solution for Two-Person Games With Biased Information Patterns," IEEE Trans. on Automatic Control, Vol. AC-17, No. 6, pp. 791-798, 1972.

[3] M. Simaan and J.B. Cruz, Jr., "On the Stackelberg Strategy in Nonzero-Sum Games," J. Optimization Theory and Applications, Vol. 11, No. 5, pp. 533-555, May 1973.

[4] M. Simaan and J.B. Cruz, Jr., "Additional Aspects of the Stackelberg Strategy in Nonzero-Sum Games," J. Optimization Theory and Applications, Vol. 11, No. 6, pp. 613-626, June 1973.

[5] D. Castanon and M. Athans, "On Stochastic Dynamic Stackelberg Strategies," Automatica, Vol. 12, pp. 177-183, 1976.

[6] D.A. Castanon and N.R. Sandell, "On Hierarchical Stackelberg Optimization Problems," CDC 1977, pp. 104-107.

[7] T. Basar and H. Selbuz, "Closed-Loop Stackelberg Strategies With Applications in the Optimal Control of Multilevel Systems," IEEE Trans. on Automatic Control, Vol. AC-24, No. 2, pp. 166-178, 1979.

[8] G.P. Papavassilopoulos and J.B. Cruz, Jr., "Nonclassical Control Problems and Stackelberg Games," IEEE Trans. on Automatic Control, Vol. AC-24, No. 2, April 1979.

[9] G.P. Papavassilopoulos and J.B. Cruz, Jr., "Sufficient Conditions for Stackelberg and Nash Strategies With Memory," to appear in J. Optimization Theory and Applications.

[10] Y.C. Ho, P.B. Luh and R. Muralidharan, "Information Structure Stackelberg Game and Incentive Controllability," submitted for publication, 1980.

[11] K. Shimizu and E. Aiyoshi, "A New Computational Method for Stackelberg and Min-Max Problems by Use of a Penalty Method," submitted for publication in IEEE-AC, 1980.

[12] J.W. Blankenship and J.E. Falk, "Infinitely Constrained Optimization Problems," J. Optimization Theory and Applications, Vol. 19, No. 2.

[13] D.G. Luenberger, "Introduction to Linear and Nonlinear Programming," Addison Wesley, 1973.

[14] E. Polak and D.Q. Mayne, "An Algorithm for Optimization Problems With Functional Inequality Constraints," IEEE Trans. on Automatic Control, Vol. AC-21, No. 2, pp. 184-193, April 1976.

ON DUALITY IN STOCHASTIC GAMES*

Rodolfo A. Milito, Rafael A. Padilla, Consuelo S. Padilla
Systems Engineering Laboratory
Venezuelan Institute for Scientific Research
IVIC, Apartado 1827, Caracas 1010 A, Venezuela

ABSTRACT

A dynamic stochastic non zero sum game is considered. The presence of parameter uncertainty confronts each player with the problem of finding a strategy with dual properties. The concepts of duality and active learning, well known in stochastic control theory, are applied to games. The innovations dual control concept is used to derive adaptive strategies for a state space description of the stochastic game. A weighting matrix quantifies the compromise between learning and immediate reward adopted by each player. Both closed-loop Nash and feedback Stackelberg games are analyzed.

I. Introduction

Many decision making situations involve the modelling and design of systems in which more than one controller is present and there is uncertainty in the parameters of the equation describing the system. To tackle this kind of problem, when a stochastic model is used for the uncertainties in the system, each decision maker has the choice of taking into account or neglecting such uncertainties in the design of its strategy and, furthermore the effect of the strategies on the uncertainty. For short horizon problems, with constant random parameters, the effect of each decision maker's strategy on the uncertainties in the system may become an important factor to be considered in the design of the strategies. In this paper we formulate an adaptive game problem and introduce the concepts of cautious, certainty equivalence, and dual /5/ strategies for non zero sum, discrete, linear, quadratic games. Both the Nash and the Stackelberg solution concepts /1-4/ are extended to this kind of problems.

In the context of stochastic adaptive control theory, the trade-off between the short term effect of the control tending to minimize the stage

---

* This work was supported in part by the National Council of Scientific and Technological Research (CONICIT) under grants DDCT-ELE-2 and S11-176 and in part by the National Science Foundation under grant INT-77-10929.

cost and the learning effect of the control that will lead to a better overall performance, has been recognized for a long time /9/. The method of considering the unknown parameters as additional states to be estimated from noisy observations allows one to consider the problem as one of stochastic nonlinear control with imperfect state measurements whose optimal solution is defined by the dynamic programming algorithm. The difficulty to implement an optimal solution, even for the most trivial problems and short horizons, is far beyond any real time computing capability. Several attempts have been made (5-7, and references therein) to capture the dual nature of the problem in the control case in a cost functional which rewards not only the short term control action but the estimation effort to guarantee a better overall performance.

One of the most recent approaches /8/ is the so called "Innovations Dual Control" (IDC). The IDC minimizes the one stage cost functional

$$J_k^{IDC}\left[u(k),\ I(k)\right] = E\{\left[y(k+1)\right]^2 - \lambda(k+1)\nu^2(k+1) \,\big|\, I(k)\}\ , \qquad (1)$$

at each stage. In (1) y(k+1) is the next output of a linear system whose output it is desired to regulate, over a finite horizon of length N, I(k) is the information available at the time the control u(k) is being selected, $\lambda(k+1)$ is a design weighting parameter, and $\nu(k+1)$ is the innovations which will be fed to the Kalman filter algorithm which is being used to estimate the unknown parameters of the system. The rationale behind (1) is that the greater the covariance of the innovations, the better the estimates to be obtained in the next stage. The resulting reduced uncertainty about the system parameters leads to a better overall performance.

As stated earlier, in this paper we formulate a linear quadratic non zero sum two-player stochastic game for a system description that contains unknown parameters. The underlying system is represented in state form with perfect state information for both players. The innovations approach introduced in /8/, is extended to consider this game case. Two different solution concepts for the game are studied: the closed loop Nash, and the feedback Stackelberg strategies. For both solution concepts dual adaptive game formulations are considered that use the innovations approach.

The paper is organized as follows. In the next section, section II, the Innovations Dual Controller is reviewed and its most important features are discussed. In section III the adaptive game problem is formulated and the Dual Nash and Stackelberg solution concepts are developed. Certainty equivalence and cautious strategies are defined as particular cases of the dual ones. Sufficient conditions are given for the existence of solutions.

In section IV a numerical example is presented that illustrate the existence of the dual effect and the need for dual strategies in this type of problems. Concluding remarks are included in section V.

II. Dual Adaptive Control. The IDC Approach

Before proceeding to the dual adaptive game formulation, let us review the derivation of the IDC /8/.

Consider a SISO stochastic linear system with constant unknown parameters:

$$y(k+1)= \phi^T(k)\Theta + e(k+1) \tag{2}$$

where

$$\phi^T(k)\overset{\Delta}{=} [u(k) \quad \psi^T(k)] \overset{\Delta}{=} [u(k)u(k-1) \ \dots \ u(k-m+1) \ y(k) \ \dots \ y(k-n+1)]$$

$$\Theta^T \overset{\Delta}{=} [b_1 \dots b_m a_1 \dots a_n] = [b_1 \ \alpha^T] \qquad E\{e_k^2\}= \sigma^2$$

$e(k+1)$: sequence of white gaussian noise $\qquad E\{e_k\}= 0$

It is desired to design a controller that minimizes the deviation of the output of the system from zero, over a finite horizon of lenght N. This control objective is quantified by the cost function:

$$J= E \sum_{k=0}^{N-1} [y(k+1)]^2 \quad , \tag{3}$$

that is to be minimized with respect to the control sequence. It is well known that the optimal control sequence is determined by Dynamic Programming. The prohibitive amount of storage and computation required by the DP algorithm prevents the use of the optimal controller in any practical situation. The IDC was proposed as an alternate, computationally feasible, suboptimal explicit way of achieving the compromise between learning and control implied by the DP algorithm. It is defined as the controller that minimizes:

$$J_k^{IDC} [u(k), I(k)]= E\{[y(k+1)]^2 - \lambda(k+1) \nu^2(k+1) \big| I_k\} \tag{4}$$

$$k= 0, 1, \dots, N-1$$

where

$$\nu(k+1)\overset{\Delta}{=} y(k+1) - \phi^T(k) \hat{\Theta}(k) \tag{5}$$

$$I_k= \{u(k-1), \dots, u(0), y(k), \dots, y(1), I(0)\}$$

The explicit control input u(k) obtained by minimizing J_k^{IDC} is given by:

$$u^{IDC}(k) = -\frac{\{[1-\lambda(k+1)]P^T_{\alpha b_1}(k)+\hat{\alpha}^T(k)\hat{b}_1(k)\}\psi(k)}{[1-\lambda(k+1)]P_{b_1}(k) + \hat{b}_1^2(k)} \tag{6}$$

The parameters estimates are obtained recursively using a Kalman filter. The covariance matrix P(k) is partioned as follows, in accordance to the partitioning of Θ in (2)

$$P(k) \triangleq \left[\begin{array}{c|c} P_{b_1}(k) & P^T_{b_1}(k) \\ \hline P_{b_1}(k) & P(k) \end{array}\right]$$

Before proceeding to extend this approach to the game situation, several remarks are in order:

a) The conflict between learning and control /8/ for the problem just posed is a finite horizon concept. When the parameter estimation error covariance becomes small enough, not much more learning is possible, and most of the effort should be devoted to control.

b) The IDC encompasses a continuum of controllers which range between the Cautious Controller ($\lambda(k+1) \equiv 0$) and the Certainty Equivalence Controller ($\lambda(k+1) \equiv 1$).

c) The problem just formulated can be regarded as the extension to the unknown parameters case of the Minimum Variance Controller. The problem has to be reformulated if the control u(k) is to be penalized. Loosely speaking, penalizing the control is equivalent to making the controller more cautious. Just adding the weighting term multiplying $u^2(k)$ to the cost functional (4) may prevent the controller from fully exploiting the knowledge it has adquired once the covariance matrix P(k) has substantially diminished. Thus, the addition of a penalty to the control action may actually prevent the potential performance improvement to be accrued from the innovations dual controller.

III. Dual Adaptive Game Formulation

III.1 Statement of the problem.

Consider a stochastic, discrete-time, non-zero-sum, two-person, finite horizon game with the system equation given by :

$$x(k+1) = Ax(k) + Bu(k) + Cv(k) + w(k), \qquad k = 0, 1, \ldots, N-1 \tag{7}$$

where

$x \in R^n$ is the state,

$u \in R^p$ is the control action exerted by player 1,

$v \in R^q$ is the control action exerted by player 2,

A, B, C are constant matrices of appropriate dimensions that contain the unknown parameters,

w(k) is white gaussian noise with zero mean and diagonal covariance matrix

$\Sigma = \mathrm{diag}\,(\sigma_1^2, \ldots, \sigma_n^2)$

The i-th player's performance index is given by

$$J_i = E\{\sum_{k=0}^{N-1} x^T(k+1)\, Q_i(k)x(k+1) + u^T(k)R_i(k)u(k) + v^T(k)S_i(k)v(k)\}, \tag{8}$$

$$i = 1, 2 .$$

We require that for every k, k= 0, ..., N-1

$$Q_i(k) \geq 0 , \quad i = 1, 2$$

$$R_1(k) > 0 , \tag{9}$$

$$S_2(k) > 0 .$$

It is assumed that at every stage k both players have access to the information state I(k) associated with the system (7), i.e.

$$I(k) = \{I(0), x(1), \ldots, x(k), u(0), \ldots, u(k-1), v(0), \ldots, v(k-1)\}, \tag{10}$$

where I(0) summarizes the initial conditions and the a priori statistics of the unknown parameters of the system.

As stated in the introduction, we are interested in both the closed loop Nash solution and the feedback Stackelberg solution concepts. The computational requirements of the dynamic programming algorithm for both of these solution concepts make it unfeasible to determine the exact solution to the game problem of (7) and (8). Following the same reasoning as in section II, the following cost functionals are proposed:

$$J_i(k) = E\{x^T(k+1)Q_i(k)x(k+1) + u^T(k)R_i(k)u(k) + v^T(k)S_i(k)v(k) \tag{11}$$
$$- \nu^T(k+1)\,\Lambda_i(k+1)\nu(k+1) \,\Big|\, I(k)\} \;,$$
$$i = 1,\,2$$

to be used at each stage k to determine u(k) and v(k) by player 1 and 2 respectively.

In equation (11) $\nu(k+1)$ represents the innovation to be obtained at the next stage, i.e.

$$\nu(k+1) = x(k+1) - E\{x(k+1) \,\Big|\, I(k)\} \;. \tag{12}$$

The determination of the player's strategies that solve (11) requires the updated mean and covariance of the unknown parameters. These are determined from standard Kalman filter equation:

$$\begin{aligned} \hat{\Theta}(k) &= \hat{\Theta}(k-1) + K(k)\nu(k) \;, \\ K(k) &= P(k-1)H^T\,(HP(k-1)H^T + \Sigma)^{-1} \;, \\ P(k) &= P(k-1) - K(k)HP(k-1) \;, \\ \nu(k) &= x(k) - H\hat{\Theta}(k-1) \;, \end{aligned} \tag{13}$$

where

$$\Theta^T = \left[a^1 b^1 c^1 \;\ldots\; a^n b^n c^n\right] \;, \tag{14}$$

a^i, b^i, c^i denote the i-th rows of A, B and C, respectively and

$$H = \begin{bmatrix} \phi^T(k-1) & 0 \;\ldots & 0 \\ \vdots & & \vdots \\ 0 \;.\;.\;.\;.\;.\;. & 0 & \phi^T(k-1) \end{bmatrix} \;, \tag{15.a}$$

$$\phi^T(k-1) = \left[x^T(k-1) \,\vdots\, u^T(k-1) \,\vdots\, v^T(k-1)\right] \;. \tag{15.b}$$

In (13) $\hat{\Theta}(k)$ is the conditional mean of given I(k) and P(k) is the conditional covariance matrix given I(k). If the parameters are initially independent by rows and the noise covariance Σ is diagonal, the filter (13) decomposes into n Kalman filters each of which describing the evolution of conditional statistics of $\Theta_i^T = [a^i\ b^i\ c^i]$.

III.2 The Dual Nash Solution Concept.

Suppose the two players decide to play a Nash strategy. The cost functionals (11) can be rewritten as

$$J_i(k) = E\{(x^T(k+1) - \hat{x}^T(k+1)(Q_i - \Delta_i)(x(k+1) - \hat{x}(k+1) \,|\, I(k)\} \tag{16}$$
$$+ \hat{x}^T(k+1)Q_i\hat{x}(k+1) + u^T(k)R_i(k)u(k) + v^T(k)S_i(k)v(k)$$

$$i = 1,\ 2$$

Sufficient conditions for $J_i(k)$, i= 1, 2, to be convex in u(k) and v(k) respectively, are

$$Q_i(k) - \Delta_i(k+1) \geq 0 \tag{17}$$
$$R_1(k) > 0 \tag{18}$$
$$S_1(k) > 0 \tag{19}$$

Recall that condition (17) is the natural extension of the condition for the control problem of a single input single output system $0 \leq \lambda \leq 1$. Notice also that $\Delta_i = Q_i$ implies Player i plays a "certainty equivalence" strategy, while $\Delta_i = 0$ implies a "cautious" strategy is followed.

Replacing (7) and (12) in (16) we obtaine:

$$\begin{aligned} J_i(k) = {} & x^T(k)L_i(k)x(k) + u^T(k)N_i(k)u(k) + v^T(k)M_i(k)v(k) \\ & + 2u^T(k)F_i(k)x(k) + 2v^T(k)G_i(k)x(k) + 2v^T(k)H_i(k)u(k) \\ & + E\{w^T(k)(Q_i(k) - \Lambda_i(k+1)w(k)\} \ , \end{aligned} \tag{20}$$

where

$$\begin{aligned} L_i &= \hat{A}^T(k)Q_i\hat{A}(k) + E\{(A - \hat{A}(k))^T(Q_i - \Delta_i)(A - \hat{A}(k))\} \ , \\ N_i &= \hat{B}^T(k)Q_i\hat{B}(k) + E\{(B - \hat{B}(k))^T(Q_i - \Delta_i)(B - \hat{B}(k))\} + R_i \ , \\ M_i &= \hat{C}^T(k)Q_i\hat{C}(k) + E\{(C - \hat{C}(k))^T(Q_i - \Delta_i)(C - \hat{C}(k))\} + S_i \ , \\ F_i &\triangleq \hat{B}^T(k)Q_i\hat{A}(k) + E\{(B - \hat{B}^T(k))^T(Q_i - \Delta)(A - \hat{A}(k))\} \ , \\ G_i &\triangleq \hat{C}^T(k)Q_i\hat{A}(k) + E\{(C - \hat{C}^T(k))(Q_i - \Delta_i)(A - \hat{A}(k))\} \ , \\ H_i &\triangleq \hat{C}^T(k)Q_i\hat{B}(k) + E\{(C - \hat{C}^T(k))(Q_i - \Delta_i)(B - \hat{B}(k))\} \ . \end{aligned} \tag{21}$$

According to conditions (17) to (19)

$$N_1 > 0 \; , \qquad M_2 > 0 \; , \tag{22}$$

the Nash strategy is simply obtained making

$$\frac{\partial J_2(k)}{\partial v(k)} = 0 \tag{23}$$

$$\frac{\partial J_1(k)}{\partial u(k)} = 0 \tag{24}$$

A necessary and sufficient condition for the Nash solution to exist is that the system of equations:

$$M_2^{(k)} v(k) + G_2(k)x(k) + H_2(k)u(k) = 0 \quad , \tag{25}$$

$$N_1(k)u(k) + F_1(k)x(k) + H_1^T(k)v(k) = 0 \quad , \tag{26}$$

have a unique solution. If this is the case the explicit solution can be computed explicitly from (25) and (26) in a straightforward manner.

III.3 The Dual Stackelberg Solution Concept.

The strategy selected is the feedback Stackelberg with player 1 as the leader and player 2 as the follower. At instant k the leader plays u(k) and this decision is available to the follower, who plays v(k).

$R_2(k) \geq 0$, $S_1(k) \geq 0$. This conditions guarantee that $J_i \geq 0$

The cost functional may again be rewritten as:

$$\begin{aligned} J_i(k) = {} & x^T(k)L_i(k)x(k) + u^T(k)N_i(k)u(k) + v^T(k)M_i(k)v(k) + \\ & 2u^T(k)F_i(k)x(k) + 2v^T(k)G_i(k)x(k) + 2v^T(k)H_i(k)u(k) + \\ & E\{w(k)(Q_i(k) - \Delta_i(k))w^T(k)\} \; . \end{aligned}$$

The definitions given is III.2 hold for this paragraph. The reaction set of player 2 is found minimizing. $J_2(k)$ with respect to v(k) for given u(k),

$$\partial J_2(k)/\partial v(k) = 0 \quad , \tag{27}$$

yielding: $M_2(k)v^*(k) + G_2(k)x(k) + H_2(k)u(k) = 0$. (28)

Player one decides on his strategy knowing that, and therefore

$$\partial J_1(u(k),\ v^*(u(k))/\partial u(k) = 0 \quad , \tag{29}$$

obtaining:

$$\left[N_1(k)+H_2^T(k)M_2^{-1}(k)M_1(k)M_2^{-1}(k)H_2(k)-\ H_2^TM_2^{-1}H_1-H_1^TM_2^{-1}H_2\right]u(k) + \left[F_1(k)+H_2^T(k)M_2^{-1}(k)M_1(k)M_2^{-1}(k)G_2(k)-H_2^T(k)M_2^{-1}(k)G_1(k)+H_1^T(k)M_2^{-1}(k)G_2(k)\right]= 0 \tag{30}$$

Necesary and sufficient conditions for the Stackelberg solution to exist is for the system of equations (28) and (30) to be uniquely solvable. For both the closed loop dual Nash solution of (25) and (26) and the feedback dual Stackelberg solution of (28) and (30), we define, as limiting cases the respective cautious strategy for player i, i=1, 2 by setting $\Lambda_i= 0$, and the certainty equivalence strategy for player i, i= 1,2 by setting $\Lambda_i= 0_i$. These definitions are suggested by the corresponding definitions in the control case /8/.

Numerical Examples

A simple example was simulated to illustrate the behaviour of dynamic games in the presence of parameter uncertainties when several degrees of compromise between caution and probing are selected by the players.

The following single output, first order system was considered:

$$y(k+1)= -0.9\ y(k) + 0.8\ u(k) + 0.7\ v(k) + e(k)$$

where e(k) is a white noise sequence with $E\{e(k)\}= 0$, $E\{e^2(k)\}= 1$. The weighting factors are: $q_1=1$, $r_1=0.01$, $s_1=0$, $q_2=1$, $r_2=0$, $s_2=0.002$.

The covariance matrix of the parameters was initialized at $P(0)=100\ I_3$. Table 1, show the average performance of 25 realizations of the process, for different values of λ_1 and λ_2. The performance is evaluated for each realization and each player i, i=1,2, by means of

$$v_i= \frac{1}{N}\sum_{k=0}^{N-1} y^2(k+1)\ q_i + u^2(k)r_i + v^2(k)s_i$$, where N is the length of the horizon.

The results shown corresponding to an adaptive Nash game over a horizon of 15 stages, for different values of λ_1 and λ_2.

General conclusions about the behavior of the strategy concepts introduced in this paper cannot be easily derived form any particular example. Nevertheless, some interesting remarks can be made.

The dual effect is noticeable and can be favorably manipulated with the proposed extension of the innovations approach. Large weights on $\mu(k)$ and $v(k)$ may dominate the dual effect. In those cases, a naive application of the Innovations Dual Strategy does not necessarily improve the resulting cost , and a Certainty Equivalence Strategy ($\Delta_i \equiv Q_i$) may result preferable. The effectiveness of the Innovations Dual Concept is obviously dependent on the evolution of the error covariance of b during the time horizon of the problem.

The combined learning effect of the two players' actions cannot be easily deduced as a direct consequence of each player's attitud towards the uncertainty. Preliminary simulation results of the adaptive dual Stackelberg game have not been conclusive in indicating improved performance due to active learning.

V. Concluding Remarks

The Innovations Dual Game solution concept for stochastic Nash and feedback Stackelberg games with uncertain parameters has been introduced. Simulations show the existence of the dual effect. These simulations reveal that due to the presence of r_1 and s_2, the average accumulated cost is not a convex function of the parameters λ_i. Further research is needed in order to quantify the potential benefits of active learning in adaptive dual games.

References

/1/ Starr, A.W. and Y.C.Ho, "Nonzero-Sum Differencial Games", JOTA, Vol.3, No.3, pp. 184-206, 1969.

/2/ Simaan, M. and J.B.Cruz, "On the Stackelberg strategy in nonzero-sum games", JOTA, Vol. 11, No. 5, May 1973, pp. 533-555.

/3/ Simaan, M. and Cruz, J.B., "Additional aspects of the Stackelberg strategy in nonzero-sum games", JOTA, Vol. 11, No.6, 1973, pp. 613-626.

/4/ Castanon, D. and M. Athans, "On stochastic dynamic Stackelberg strategies", Automatica, Vol. 12, 1976, pp. 177-183.

/5/ Wittenmark, B., "Stochastic adaptive control methods: a survey", Int. J. Control, 1975, Vol. 21, No.5, pp. 705-730.

/6/ Padilla, C.S. J.B.Cruz, and R.A. Padilla, "Conceptual framework of SAFER control", 18th. IEEE C.D.C., Dec. 1978, pp.187-190, Fort Lauderdale.

/7/ Padilla, C.S. and J.R.Cruz, Jr. "Sensitivity adaptive feedback with estimation redistribution", IEEE Trans.on Aut.Control, Vol.AC-23,June 78 .

/8/ Milito,R, C.S. Padilla, R.A. Padilla, D. Cadorín, "Dual control through innovations", 19th. IEEE Conf.on Dec. and Control, Dec.1980, Albuquerque.

/9/ Feldbaum, A.A., "Dual control theory I-IV", Automation and Remote Control", 1960, Vol.21, pp.1240-1464, Vol.-22,1961, Vol.22, p.3-16, 129-143.

λ_1 ↓						V_1 V_2
1.	6.21 6.04	26.92 25.92	48.54 47.60	4.98 4.90	33.79 33.84	
0.75	3.66 3.66	3.44 3.44	3.66 3.65	2.72 2.72	10.69 10.79	
0.50	3.46 3.46	3.63 3.63	4.22 4.21	4.04 4.04	4.09 4.11	
0.25	5.90 5.90	3.35 3.35	3.10 3.10	3.68 3.69	29.92 30.14	
0.	4.36 4.36	4.39 4.39	3.80 3.80	2.98 2.98	37.32 37.51	
	0.	0.25	0.50	0.74	1.	→ λ_2

TABLE 1:

Nash solution-time horizon N= 15.

PRODUCT FORM AND OPTIMAL LOCAL FEEDBACK FOR A MULTIINDEX MARKOV CHAIN *

J.P. Quadrat - M. Viot
I N R I A
Domaine de Voluceau
B.P. 105 - Rocquencourt
78150 LE CHESNAY

ABSTRACT For a network of queues of Jackson type, with derivation from the entry to the exit when the capacity of storage of a mean is saturated, we prove the product form property for the equilibrium probability law. We give an algorithm to compute the optimal local feedback for the controled system in the case where the control is the service rate of each individual queue. We prove the convergence of the algorithm to a strategy optimal player by player.

INTRODUCTION

When we want to compute the optimal feedback for a large scale stochastic control problem we meet the "curse of dimensionality" difficulty which indicates that the number of parameters to compute grows exponentially with the dimension of the system. One way to avoid this difficulty is to limit our ambition and compute the optimum only for a subclass of feedbacks for example for the class of local feedbacks. It appears very quickly that in general this last problem is more difficult than the former one. But for some particular systems there is a gain to impose this limitation on the admissible strategies. For example when the dynamic of the system is decoupled, with perhaps a coupling criterion we can see that a player by player optimum strategy is reachable when the global optimum is not computable. The purpose of this paper is to generalize this situation to system for which the equilibrium probability has the product form, that is this equilibrium probability can be written as a product of independant marginal laws of probability. In the first part we describe some finite valued Markov chain with this property. They are of Jackson type (Jackson [1], B.C.M.P. [2], Kelly [3], F.K.A.S [4], Schassberger [617]) with a derivation of the input of customer,to the output whenever the queue is full. Then we suppose that the output of each queue is a control variable. We give a necessary and sufficient condition for a control to be optimum player by player where a player is a server of a queue. And we give an iteration policy algorithm (Howard [8], Miller,Veinott [7]) to compute this control.

I. A FINITE VALUED MARKOV CHAIN WITH THE PRODUCT FORM PROPERTY (Fig. 2)

We consider a set of interconnected queues, with finite capacities, and derivation of the input to the output when a queue is full. We suppose that the output rate is exponentially distributed and that the input of the system depends of the total number of customers. This system is of Jackson type with derivation. First we give a precise model and then give the equilibrium probability law.

I.1. The model

We denote by $E = \{1, 2, \dots, m\}$ the set of the index of the queues. The index $\{o\}$ represents the exterior considered as a special queue and $\bar{E} = E \cup \{o\}$.

If n_i, $i \in E$ is the number of customers in the queue i we denote by $n(n_o = \sum_{i=1}^{m} n_i, n_1, \dots, n_m)$. $\check{n}_i$, $i \in E$ is the capacity of the queue i. We use the notations $\check{n} = (\check{n}_o = \sum_i \check{n}_i, \check{n}_1, \check{n}_2, \dots, \check{n}_m)$ and $N = \{n, o \le n_i \le \check{n}_i, i \in E\}$.

The service rate is denoted by μ:

(i) $\mu_i : [o, \check{n}_i] \to \mathbb{R}^+$, $i \in \bar{E}$.

We suppose that $\mu_i(n_i) \ge \mu \;\; \forall i, \forall n \in N$, and $\mu_i(o) = o$. $\forall i \in E, \mu_o(\check{n}_o) = o$.

We call r the routage matrix :

(ii) $r : \underset{i}{\bar{E}} \times \underset{j}{\bar{E}} \to \underset{r_{ij}}{[0, 1]}$ r, being a stochastic matrix associated to an irreducible markov chain. r_{ij} indicates the proportion of customers leaving the queue i going into the queue j. r_{oj} is the proportion of the input going into the queue j and r_{jo} the proportion of the output of the queue j leaving the system.

When a queue is full the input of the queue is derived to the output of the queue and the corresponding customer should been trying to enter in a queue until it finds a place free. If $F(n) \subset E$, is a set of queues which have a place free, when the state of the system is n, we can describe this phenomenon using the routage matrix r^F (r^F_{ij} = the probability starting from a queue belonging to $i \in E$ ending in the queue $j \in F$ (the first queue with a free place))

$r^F : \underset{i}{\bar{E}} \times \underset{j}{F} \to \underset{r^F_{ij}}{[0,1]}$.

Using the notations : $\bar{F} = F \cup \{o\}$, $F^c = C_E^F$, $r^F = \begin{array}{c|c} & \bar{F} \\ \hline \bar{F} & r_1^F \\ F^c & r_2^F \end{array}$ $\quad r = \begin{array}{c|cc} & \bar{F} & F^c \\ \hline \bar{F} & r_1 & r_2 \\ F^c & r_3 & r_4 \end{array}$

$s = \sum_{i=o}^{+\infty} r_4^i$.

We have : $r_1^F = r_1 + r_2 s r_3$, $r_2^F = s r_3$.

We remark that s is meaningfull because the chain is irreductible, so there is a path connecting F^c to F, therefore r_4 has no eigenvalue of modulus 1.

We need the introduction of the following operators :

$T_{ij} : N \to \bar{N}$

$(n_o, n_1, \dots, n_m) \to T_{ij}(n)$

with

$T_{ij}(n) = (n_o, n_1, \ldots, n_i-1, \ldots, n_j+1, \ldots, n_m)$ for $i \in E$, $j \in E$;
$T_{io}(n) = (n_o-1, n_1, \ldots, n_i-1, \ldots, n_m)$ for $i \in E$;
$T_{oi}(n) = (n_o+1, n_1, \ldots, n_i+1, \ldots, n_m)$ for $i \in E$.

and

$$\bar{N} = \{(n_o, n_1, \ldots, n_m) \mid n_o = \sum_{i>o} n_i,\ -1 \le n_i \le \check{n}_i+1,\ i \in E\}$$

Now it is possible to define the transition rates of the system denoted by q

$$q : N \times N \to \mathbb{R}^+$$
$$(n,n') \quad q(n, n').$$

The non zero terms of q are :

$$q(n, T_{ij}n) = \mu_i(n_i)\, r_{ij}^{F_i(n)}, \text{ for } i \in E,\ j \in F_i(n),\ n \in N,$$
$$\text{with } F_i(n) = \bar{F}(n) \cup \{i\}.$$

I.2. A product form result

The equilibrium probability p(n) satisfies the global balance equation

$$(1.1) \qquad \sum_{i\in E, j\in E} p(T_{ij}^{-1}n)\, q(T_{ij}^{-1}n, n) = p(n) \sum_{i\in \bar{E}, j\in \bar{E}} q(n, T_{ij}n).$$

We remark that $T_{ij}^{-1} = T_{ji}$ for $j \in E$, $i \in E$; and because $T_{oi}T_{io} = T_{io}T_{oi} = I$ we have $T_{oi}^{-1} = T_{io}$ and $T_{io}^{-1} = T_{oi}$, so (1.1) can be written also as

$$(1.1') \qquad \sum_{i\in \bar{E}, j\in \bar{E}} p(T_{ij}\, n)\, q(T_{ij}\, n, n) = p(n) \sum_{i\in \bar{E}, j\in \bar{E}} q(n, T_{ij}n)$$

Theorem 1 The system described in I.1 admits a unique invariant measure of probability

$$(1.2) \qquad p(n) = C \prod_{i=1}^{m} \left[\prod_{k=1}^{n_j} (e_i/\mu_i(k))\right] \prod_{k=o}^{n_o-1} (\mu_o(k)/e_o), \quad n \in N \text{ where}$$

the vector e satisfies

$$(1.3) \qquad e\, r = e$$

and C is a constant of normalization

Proof
We verify that (1.2) satisfies the following partial balance equation

$$(1.4) \qquad p(n) \sum_{j\in \bar{E}, j\neq i} q(n, T_{ij}n) = \sum_{j\in \bar{E}, j\neq i} p(T_{ij}n)\, q(T_{ij}n, n),\ i \in E.$$

Let us first verify (1.4) for $o \le n < \check{n}$. In this case

$q(n, T_{ij}n) = \mu_i(n_i)\, r_{ij}$
$q(T_{ij}n, n) = \mu_j(n_j+1)\, r_{ji}$, $j \in E$
$q(T_{io}n, n) = \mu_o(n_o-1)\, r_{oi}$.

But $p(T_{ij}n)/p(n) = (\mu_i(n_i)/e_i)(e_j/\mu_j(n_j+1))$ for $j \in E$, $j \neq i$;

$p(T_{io}n)/p(n) = (e_o/\mu_o(n_o-1))(\mu_i(n_i)/e_i)$ and using (1.3)(1.4) is easily verified.

Let us verify (1.4) when $C_E F(n) \neq \emptyset$. In this case we have :

$$q(n,T_{ij}n) = \mu_i(n_i) r_{ij}^{F_i(n)} \; 1^{(j)}_{F_i(n)}, \text{ for } j \in E \text{ ;}$$

$$q(T_{ij}n,n) = \mu_j(n_j+1)\; r_{ji}^{F_i(n)} \; 1^{(j)}_{F_i(n)} \text{ because } F_j(T_{ij}n) = F(n) \cup \{i\} \text{ when}$$

$$j \in F_i(n) \text{ ;}$$

$$q(T_{io}n,n) = \mu_o(n_o-1)\; r_{oi}^{F_i(n)} \; 1^{(o)}_{F_i(n)}, \text{ because } o \text{ always belongs to } F_i(n).$$

Therefore, the proof is the same as the case $0 \leqslant n < \check{n}$ provides we con prove that

$$(1.5) \qquad \sum_{j \in F_i(n)} e_j \; r_{ji}^{F_i(n)} = e_i .$$

Using the notation $e_{F_i} = \{e_k,\ k \in F_i\}$ $\quad e_{F_i^c} = \{e_k,\ k \in F_i^c\}$ from (1.3) we have :

$$e_{F_i} r_1 + e_{F_i^c} r_2 = e_{F_i^c}$$

$$e_{F_i} r_3 + e_{F_i^c} r_4 = e_{F_i}$$

which prooves that

$$e_{F_i} r_1 + e_{F_i} r_3 s r_2 = e_{F_i} \quad \text{which is (1.5).}$$

Therefore p(n) is an equilibrium probability. The uniqueness comes from (i) and (ii) which guarantees that it is possible to reach every point of $\mathcal{N}$ from every other point.

From the theorem the following result is easily deduced.

Corollary. The invariant measure of the system described in (1.1) $p(n_1, \ldots, n_m) = C \prod_{i=1}^{m} p_i(n_i)\; p_o(\sum_{i \in E} n_i - 1)$ where $p_i(n_i)$, $i \in E$ is the invariant measure of the queue i, in isolation with input $e_i 1^{(n_i)}_{n_i < \check{n}_i}$ and $p_o(n_o)$ is the invariant system measure of an aggregate queue, describing the total number of customers in the system with input $\mu_o(n_o)$ and output $e_o 1^{(n_o)}_{n_o} {>}0 (n_o)$.

Remark This kind of boundary behaviour is not exactly the truncation described in Kelly [3], but it preserves also the partial balance condition and accordingly the product form property.

II. OPTIMAL LOCAL FEEDBACK OF JACKSONIAN QUEUES

II.1. The control problem

Now we need to make one more hypothesis.We assume that:

(iii) μ_o is independant of n_o $\quad N = \prod_{i=1}^{m} \{o, \check{n}_i\}$.

Given the network of queues described in I.1,(where now the service rates μ are control variables) and a cost function c :

$c : N \times [\hat{\mu}, \check{\mu}]^m \to \mathbb{R}^+$ continuous, we want to optimize the objective function:

$$(2.1) \qquad J(\mu) = \lim_{T\to\infty} 1/T \sum_{t=o}^{T-1} c(n(t), \mu(t))$$

in the class of local feedback S_L that is :

$$S_L = \{\mu \mid \mu_i : \{1, \check{n}_i\} \to U \subset [\hat{\mu}, \check{\mu}]\} .$$

In this paragraph we take advantage of the product form property to characterize strategies belonging to S_L, which are optimal player by player (i.e. each μ_i is one of these players). An equivalent formulation is :

$$(2.1.') \qquad \min_{\mu \in S_L} J(u) = \sum_{n \in N} c(n, \mu(n))\, p_\mu(n)$$

where $p_\mu(n)$ is given by (1.1), because $\forall \mu \in S_L$ the system described in I.1 has a unique invariant measure $p_\mu(n)$.

Let us introduce the following notations:

$g(n_i) = e_i^{(n_i)} 1_{n_i < \check{n}_i}$ where e_i is defined by (1.3). $\forall i \in E$, $A_i(\mu_i)$ is a matrix characterized by :

$$A_i(\mu_i)_{k,k+1} = g_i(k) \quad \forall k,\ o \le k \le \check{n}_i$$
$$A_i(\mu_i)_{k,k-1} = \mu_i(k)$$
$$A_i(\mu_i)_{k,k} = -g_i(k) - \mu_i(k)$$
$$A_i(\mu_i)_{k,\ell} = o \text{ elsewhere}$$

$A*$ is the transpose of A,

$W \in \mathbb{R}^+$ shall be interpreted as the optimal cost,

$V_i : [o, \check{n}_i] \to R$ could be interpreted as a Bellman function.

The next result is the characterization of a Nash point $\mu^* \in S_L$ that is μ^* which satisfies :

$$J(\mu_1^*, \mu_2^*, \ldots, \mu_m^*) \le J(\mu_1^*, \mu_2^*, \ldots, \mu_i, \ldots, \mu_m^*) \quad \forall \mu_i : \{1, \ldots, \check{n}_i\} \to [\hat{\mu}, \check{\mu}].$$

Under differentiability properties for $\mu \in S_L \to J(\mu) \in \mathbb{R}$, such a point is a local minimum.Under convexity and differentiability properties for J such a point is a global minimum.

II.2. Characterization of Nash-optimal strategies

Theorem 2. A necessary and sufficient condition for W to be a Nash optimal cost is that $(\mu_i^*, p_i, V_i, W, i \in E)$ satisfying

$$(2.2) \qquad A_i^*(\bar{\mu}_i^*)p_i = 0, \quad \sum_{k=o}^{n_i} p_i(k) = 1$$

$$(2.3) \qquad \operatorname*{Min}_{\mu_i} \{A_i(\mu_i)\, V_i + c_i(\mu_i)\} = W$$

$$(2.4) \qquad \mu_i^* \in \operatorname*{Arg\,Min}_{\mu_i} \{A_i(\mu_i)\, V_i + c_i(\mu_i)\}$$

where $c_i : \{0, \dots, \check{n}_i\} \times [\hat{\mu},\check{\mu}] \to \mathbb{R}^+$

$$k \qquad\qquad \nu \qquad c_i(k,\nu)$$

$$(2.5) \qquad c_i(k,\nu) = \mathbb{E}_p\{c(n,\mu(n)) \mid n_i = k,\ \mu_i(n_i) = \nu,\ \mu_j = \mu_j^*,\ j \neq i\} \quad \text{and}$$

$$p = \prod_{i=1}^{m} p_i, \text{ exists.}$$

Proof The invariant measure $p(\mu)$ associated to the strategy $\mu \in S_L$ can be written as $p^\mu = \prod_{j=1}^{m} p_i^{\mu_i}$ where $p_i^{\mu_i}$ is the solution of $A_i^*(\mu_i)p_i^{\mu_i} = o$, $\sum_{k=o}^{\check{n}_i} p_i^{\mu_i}(k) = 1$.

The criterium that we have to optimize can be written as:

$$\operatorname*{Min}_{\mu} J(\mu) = \sum_{n} c(n,\mu(n)) \prod_{i=1}^{m} p_i^{\mu_i}(n_i).$$

Now if μ^* is a Nash point and if we denote by $S^i_{\mu_i}$ the strategy $(\mu_1^*, \mu_2^*, \dots, \mu_i, \mu_{i+1}^*, \dots, \mu_m^*)$

$$(2.6) \qquad J(\mu^*) \leq J(S^i_{\mu i}) \quad \forall \mu_i : [0,\check{n}_i] \to [\hat{\mu},\check{\mu}]$$

then (2.6) can be written as:

$$(2.7) \qquad J(\mu^*) = \operatorname*{Min}_{\mu_i} \mathbb{E}_{p_i^{\mu_i}}\, c_i(n_i,\mu_i)$$

$$\text{with } c_i(n_i,\mu_i) = \sum_{\{n \mid n_i \text{ given}\}} c(n,\mu(n)) \prod_{j \neq i} p_j^{\mu_j^*}(n_j)$$

and because the n_i is an irreducible Markov chain $\forall \mu_i$, using the undiscounted cost dynamic programming theory, a necessary and sufficient condition for $W = J(\mu^*)$ to be the minimum cost is that it satisfies :

$$\operatorname*{Min}_{\mu_i} \{A(\mu_i)\, V_i + c_i(\mu_i)\} = W, \ \forall i.$$

The existence of a Nash point is guaranteed by the continuity of the function $\mu_i \rightarrow p_i^{\mu_i}$ (which a consequence of the continuity of the unique eigen vector associated to the eigen value o of the operator $A_i(\mu_i)$ Kato [20]. The uniqueness of p_i follows from the fact that theMarkov chain $n_i(t)$ has only one recurrent class).

The function $\mu \rightarrow c(\mu)$ is continuous, the admissible set of strategies is compact by hypothesis. Thus the optimal cost in the class of local feedbacks, which is a Nash point, exists.

II.3. A relaxation algorithm to compute the Nash optimal strategy

To compute $(\mu_i^*, p_i, V_i, W, i \in E)$ solution of (2.2) to (2.5) we can use the following policy improvement algorithm :

1) start with $\mu \in S_L$, compute p^μ, put $j = n = 1$.

2) compute $c_j(\mu_j)$
compute by a classical improvement algorithm V_j, W^n, μ_j^* the solution of
$$\underset{\mu_j}{\mathrm{Min}} \{A_j(\mu_j) V_j + c_j(\mu_j)\} = W^n , \quad \mu_j^* \in \underset{\mu_j}{\mathrm{ArgMin}} \{A_j(\mu_j) V_j + c_j(\mu_j)\}.$$

3) μ becomes $(\mu_1, \mu_2, \ldots, \mu_j^*, \mu_{j+1}, \ldots, \mu_m)$, compute $p_j^{\mu_j^*}$.

4) n becomes n+1, j=j+1 modulo n, and return to 2) until W^n converges.

Clearly the sequence W^n is decreasing and bounded by o from below, thus it converges to W^*, butwe are not assured that W^* is Nash optimal cost, excepte when the set of control is finited valued Sandell [12]. Powell's counter example [13], for example, shows the existence of limit cycles. To avoid this difficulty we can modify the step (3) which changes to (3') :

μ becomes $(\mu_1, \mu_2, \ldots, \mu_j^*, \mu_{j+1}, \ldots, \mu_m)$ only if $W^{n-1} - W^n > \varepsilon$ where ε is a given number.

Then clearly after a finite number of steps we obtain a point μ^ε which is ε-Nash optimal that is

$$J(\mu^\varepsilon) \leq J(\mu_1^\varepsilon, \ldots, \mu_{i-1}^\varepsilon, \nu, \mu_{i+1}^\varepsilon, \ldots, \mu_m^\varepsilon) + \varepsilon, \forall \nu \in [o, \check{n}_i] \rightarrow [\hat{\mu}, \check{\mu}] .$$

Now having obtained μ^ε, ε becomes $\varepsilon/2$, and we reiterate using the same algorithm with initial condition μ^ε to obtain $\mu^{\varepsilon/2}$, and so on. Clearly a subsequence of $\{\mu^{\varepsilon/2^n}, n \in N\}$ is convergent and converges to a Nash point, following the continuity of $\mu \rightarrow J(\mu)$. Nevertheless this new algorithm cannot be implemeted on a computer because the step 2 needs an ∞ number of iterations to be achieved. So we have to solve the step 2 only approximatively if we want to avoid this new difficulty.

Before giving an convergent algorithm let us introduce the notations

$$J_j : S_L \rightarrow \mathbb{R}^+ \{o, \ldots, \check{n}_j\} \times [\hat{\mu}, \check{\mu}]$$
$$(\mu_1, \ldots, \mu_m) \quad C_j \text{ defined by (2.5)}$$

$$SL_{j,\mu} : [\hat{\mu},\check{\mu}]^{\{o,\dots,\check{n}_j\}} \to \mathbb{R}^{\check{n}_j+2}$$

$$\mu_j \qquad (V_j, W) \text{ a solution of } A_j(\mu_j)\, V_j + c_j(\mu_j) = W$$

with $c_j = J_j(\mu)$.

A consequence of (i) and (ii) is that W is uniquely defined.

$$M_{j,\mu} : \mathbb{R}^{\check{n}_j+1} \longrightarrow [\hat{\mu},\check{\mu}]^{\{o,\dots,n_j\}}$$

$$V_j \qquad \mu_j$$

$$\mu_j \in \operatorname*{Arg\,Min}_{\mu} A_j(\mu)\, V_j + c_j(\mu)$$

Fig. 1 gives now the flowchart of the algorithm in which appears two supplementary indices appear, ε which was previously defined , h which is an index for an iteration of the Howard algorithm. μ, V and W are indexed by this two variables.

'Test' is a variable which counts the number of players which cannot improve their cost more than an ε, by an iteration of the Howard algorithm.

Theorem 3 $\{W_\varepsilon \mid \varepsilon = 1/2^n\}$ is a positive decreasing sequence converging to W^* which is cost optimal player by player.

From $\{\mu_\varepsilon \mid \varepsilon = 1/2^n\}$ we can extract a subsequence which is convergent, and all convergent subsequences converge to a policy optimal player by player of cost W^*.

Proof $W_\varepsilon = J(\mu_\varepsilon)$ and so it is always positive. Each time we go through the point G of the flowchart W decreases by ε. So in a finite number of iteration we certainly go through the point F of the flowchart. At this point, whatever the player, it is impossible to improve the cost by ε. So at this point the following relations are verified:

$$(2.6) \qquad A_j(\mu_{j,\varepsilon})\, V_{j,\varepsilon} + c_j(\mu_{j,\varepsilon}) = W_\varepsilon \qquad \forall\, j \in E$$

$$(2.7) \qquad A_j(\mu'_{j,\varepsilon})\, V'_{j,\varepsilon} + c_j(\mu'_{j,\varepsilon}) = W'_\varepsilon$$

$$(2.8) \qquad W'_\varepsilon \geq W_\varepsilon - \varepsilon$$

$$(2.9) \qquad A_j(\mu'_{j,\varepsilon})\, V_{j,\varepsilon} + c_j(\mu'_{j,\varepsilon}) \leq A_j(\mu_j)\, V_{j,\varepsilon} + c_j(\mu_j)$$

$$\forall\, \mu_j : [o,\check{n}_j] \to [\hat{\mu},\check{\mu}]$$

Denoting by : $A'_{j,\varepsilon} = A(\mu'_{j,\varepsilon}),\ c'_{j,\varepsilon} = c(\mu'_{j,\varepsilon})$

from (2.6) we have $A'_{j,\varepsilon}\, V_{j,\varepsilon} + c'_{j,\varepsilon} + (A_{j,\varepsilon} - A'_{j,\varepsilon})V_{j,\varepsilon} + c_{j,\varepsilon} - c'_{j,\varepsilon} = W_\varepsilon$,

which can be written as :

$$(2.10) \qquad \operatorname*{Min}_{\mu} \{A_j(\mu)\, V_{j,\varepsilon} + c_j(\mu)\} + \xi_{j,\varepsilon} = W_\varepsilon$$

with

$$\xi_{j,\varepsilon} = (A_{j,\varepsilon} - A'_{j,\varepsilon}) V_{j,\varepsilon} + c_{j,\varepsilon} - c'_{j,\varepsilon} \geq 0$$

Let us now give an estimate for $\sup_{n_j} \xi_{j,}(n_j)$.

From (2.6),(2.7):

$$A'_j(V_{j,\varepsilon} - V'_{j,\varepsilon}) + \xi_{j,\varepsilon} = W_\varepsilon - W'_\varepsilon$$

and so using (2.8) we have $W_\varepsilon - W'_\varepsilon = \sum_{n_j} \xi_{j,\varepsilon}(n_j)\, p_j^{\mu'_{j,\varepsilon}}(n_j) \leq \varepsilon$

which implies that

(2.11) $$\sup_{n_j} \xi_{j,\varepsilon}(n_j) \leq \varepsilon / \operatorname{Inf}_{n_j} p_j^{\mu'_{j,\varepsilon}}(n_j)$$

but because (i) and (ii)

$\exists\, \alpha > o \; ! \operatorname{Inf}_{j,\mu,n_j} p_j^{\mu_j}(n_j) \geq 1/\alpha$ (the hypotheses (i) and (ii) were introduced for this reason).

Thus we have obtained that:

(2.12) $$\left|\operatorname{Min}_{\mu} \{A_j(\mu) V_{j,\varepsilon} + c_j(\mu)\} - W_\varepsilon\right| \leq k\varepsilon.$$

Now because $\mu'_{j,\varepsilon}$ belongs to a compact set we can extract a subsequence of $\{\mu'_{j,\varepsilon} \mid \varepsilon = 1/2^q\}$ which is convergent. The index of this subsequence is denoted by q.

Because of (i) and (ii) $V_{j,\varepsilon}$ for all j and ,we can always choose $V_{j,}$ in such away that it is uniformly bounded and so we can extract a subsequence of $(V_{j,\varepsilon} \mid \varepsilon = 1/2^q)$ which is convergent. The corresponding subsequence is once more indexed by q.

The sequence $(W_\varepsilon \mid \varepsilon = 1/2^q,\ q \in \mathbb{N})$ is convergent and then using the continuity property of $\mu_j \to A_j(\mu_j)$, $\mu \to c_j(\mu_j)$ and the fact that $1/2^q$ converges to zero, (2.12) and the definition of $\mu'_{j,\varepsilon}$ gives the equation (2.2) to (2.5) and we have the result.

Remark This policy algorithm gives a constructive existence result of a solution of (2.2) to (2.5).

CONCLUSION

In this paper we have proved that a network of queues of Jackson type with finite capacity of storage and derivation of the input to the output (when there is saturation of this storage capacity) has the product form property. Then we have studied the control problem where the control variables are the output rates as a feedback on the number of customers waiting in the corresponding queue. We have given a necessary and sufficient condition for a strategy to be optimal player by player and an algorithm to compute this strategy. The extension to more general systems of queues for example B.C.M.P. [2] queues is currently under investigation.

REFERENCES

[1] J.R. JACKSON (1963) Jobshop like queuing system, Mgmt science 10, 131-142.

[2] F. BASKETT, M. CHANDY, R. MUNTZ, J. PALACIOS (1975),Open closed and mixed network of queues with different class of customers, J.A.C.M 22, 248-260.

[3] F.P. KELLY (1979), Reversibility and stochastic Networks, J. Wiley & Sons.

[4] P. FRANKEN, D. KONIG, V. ARNDT, V. SCHMIDT (1979), Queues and point process, Akademic-Verlag, Berlin.

[5] G. SECCO SUARDO (1978), Optimization of a closed network of queues. Report M.I.T. Electronic Systems Laboratory 02139.

[6] R.S. SHASSBERGER (1978), Insensitivity of steady state distributions of generalized semi Markov processes with speeds. Adv. Appl. Prob. 10, 836-851.

[7] B.L. MILLER, A.F. VEINOTT (1969), Discrete programming with small interest rate. An. Math. Stat. 40(2).

[8] R.A. HOWARD (1960), Dynamic programming and Markov processes. Wiley.

[9] J.G. KEMENY , J.L. SNELL, KNAP (1976), Finite Markov chains, Springer-Verlag.

[10] J.P. QUADRAT, M. VIOT, Control of diffusion processes having the product form property in the class of local Feedback, to appear.

[11] S.M. ROSS (1969), Applied probability models with optimization applications. Holdan day.

[12] N. SANDEL (1974), Control of finite state finite memory stochastic systems. M.I.T. Thesis Electronic Systems Laboratory.

[13] M.J.P. POWELL (1973), On search direction for minimization algorithms. Mathematical programming 4, 193-201.

[14] A. EPHREMIDES, P. VARAIYA, J. WALRAND (1978), A simple dynamic routine problem. University of California, Berkeley report 94720.

[15] M. REISER (1979), A queueing network analysis of computer communication networks with window flow control. IEEE Transaction on communications, 8, 1199-1209.

[16] E. GELEMBE, R.R. MUNTZ (1976), Probabilistic models of computer systems Acta Informatica 7, 35-60.

[17] R. SCHASSBERGER (1979), The insensitivity of stationary probabilistic in networks of queues. Adv. Ap. Prob. 10, 906-912.

[18] E. POLAK (1971), Computational methods in optimization a unified. Approach Academic-Press.

[19] P.P. BERTSEKAS (1976), Dynamic programming and stochastic control. Academic press.

[20] T. KATO (1966), Perturbation of linear operators. Springer Verlag.

The same kind of result can be obtained for a system of diffusion processes QUADRAT-VIOT [10].

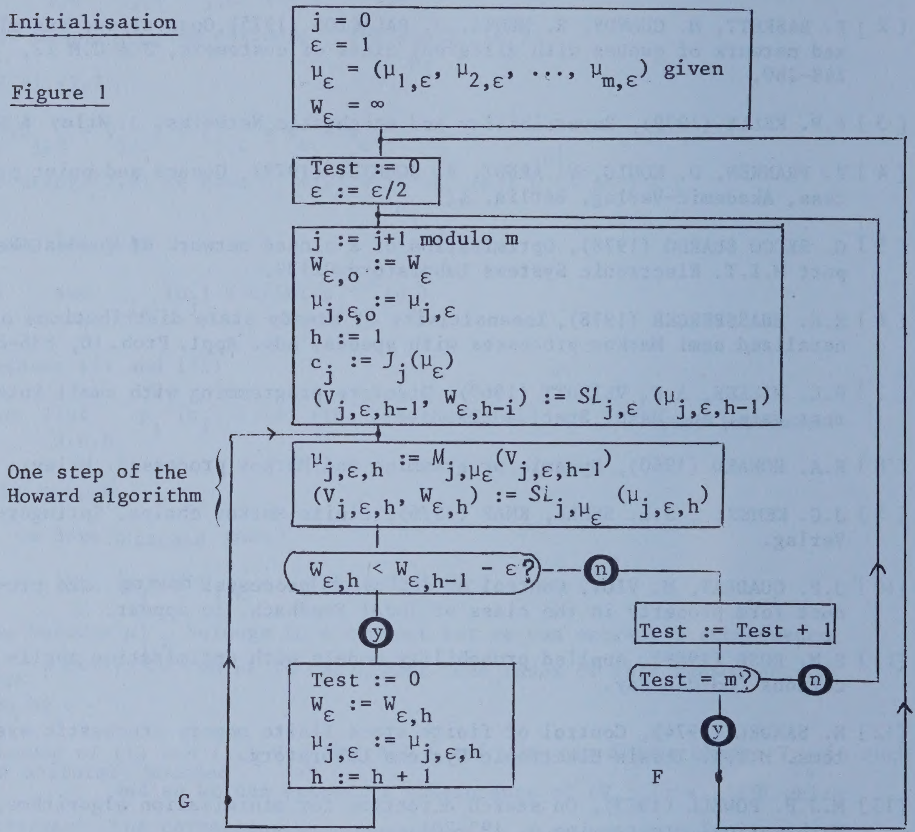

Initialisation
$j = 0$
$\varepsilon = 1$
$\mu_\varepsilon = (\mu_{1,\varepsilon}, \mu_{2,\varepsilon}, \ldots, \mu_{m,\varepsilon})$ given
$W_\varepsilon = \infty$
Test := 0
$\varepsilon := \varepsilon/2$
$j := j+1$ modulo m
$W_{\varepsilon,o} := W_\varepsilon$
$\mu_{j,\varepsilon,o} := \mu_{j,\varepsilon}$
$h := 1$
$c_j := J_j(\mu_\varepsilon)$
$(V_{j,\varepsilon,h-1}, W_{\varepsilon,h-i}) := SL_{j,\varepsilon}(\mu_{j,\varepsilon,h-1})$
One step of the Howard algorithm
$\mu_{j,\varepsilon,h} := M_{j,\mu_\varepsilon}(V_{j,\varepsilon,h-1})$
$(V_{j,\varepsilon,h}, W_{\varepsilon,h}) := SL_{j,\mu_\varepsilon}(\mu_{j,\varepsilon,h})$
$W_{\varepsilon,h} < W_{\varepsilon,h-1} - \varepsilon$?
n
y
Test := 0
$W_\varepsilon := W_{\varepsilon,h}$
$\mu_{j,\varepsilon} := \mu_{j,\varepsilon,h}$
$h := h + 1$
Test := Test + 1
Test = m?
n
y
F
G

Figure 1

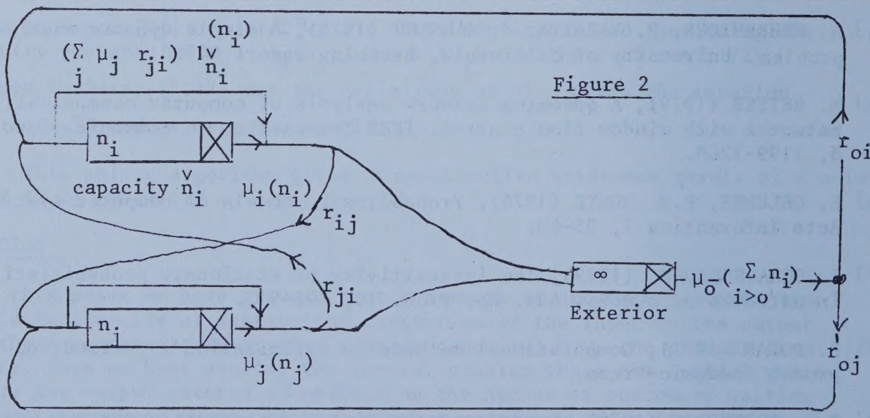

(n_i)
$(\Sigma_j \mu_j r_{ji}) 1 \vee \overset{\vee}{n}_i$
n_i
capacity $\overset{\vee}{n}_i$
$\mu_i(n_i)$
r_{ij}
r_{ji}
n_j
$\mu_j(n_j)$
∞
Exterior
$\mu_o(\sum_{i>o} n_i)$
r_{oi}
r_{oj}

Figure 2

---

[*] Supported in part by Contract 01-80RA-50154 with U.S. DOE, Office of Electric Energy Systems, and by CNRS (France) ATP-4008.

NESTED OPTIMIZATION OF WEAKLY COUPLED MARKOV CHAINS*

S. HAROLD JAVID
Systems Control Inc., Palo Alto, CA 94304

ABSTRACT

This paper develops a nested optimization technique which is based on the two time-scale properties of weakly coupled Markov chains.

1. INTRODUCTION

The properties of weakly coupled Markov chains have been invesigated in some detail [1-3]. For the discounted cost problem, work on decomposition of the optimization has produced results [1]. This paper briefly summarizes some important properties of weakly coupled Markov chains and, based on these properties, presents a nested optimization algorithm for the problem of minimizing the expected cost per unit time. This technique is based on an algorithm due to Varaiya [4,5]. It is important because it removes stiffness present in the problem of minimizing the expected cost per unit time with respect to a weakly coupled Markov chain by reducing the larger (original) problem into a number of smaller subproblems. The removal of stiffness and reduction in problem order can prove of great value since in many applications the number of states in a Markov chain can be very large (on the order of thousands for hydroscheduling [6,7]).

The problem treated here is that of minimizing the average cost per unit time

$$J(u) = \lim_{T\to\infty} \frac{1}{T+1} E \sum_{0}^{T} g(x_t, u(x_t)) \qquad (1)$$

where $x_t \in \{1,2,\ldots,s\}$ are the states of a finite state weakly coupled Markov chain whose transition probabilities at time t $(t = 0,1,\ldots)$ are dependent on the control $u(x_t)$ chosen at t. The vector p(t), whose

---

*This work was supported in part under DOE Contract DE-AC01-79-ET29243.

elements $p_i(t)$ are the probabilities of occupying the states $i = 1,2,\ldots,s$, satisfies

$$p(t + 1) = p(t)[A(u) + \varepsilon B(u)] \tag{2}$$

where $A(u)$ is block diagonal,

$$A(u) = \text{diag}[A_{11}(u), \ldots, A_{NN}(u)], \tag{3}$$

the elements $a_{ij}(u)$ and $b_{ij}(u)$ of $A(u)$ and $B(u)$ satisfy the conditions

$$a_{ij}(u) + \varepsilon b_{ij}(u) \geq 0, \quad a_{ij}(u) \geq 0 \qquad i,j=1,\ldots,s \tag{4}$$

$$\sum_{j=1}^{s} a_{ij}(u) = 1 \qquad i=1,\ldots,s \tag{5}$$

$$\sum_{j=1}^{s} b_{ij}(u) = 0 \qquad i=1,\ldots,s \tag{6}$$

and ε is a small positive parameter representing the weak coupling. The probability of transitioning from state i to state j is thus

$$P(x_{t+1} = j \,|\, x_t = i) = a_{ij}(u(i)) + \varepsilon b_{ij}(u(i)) \tag{7}$$

where the control $u(i)$ is constrained $u(i) \varepsilon\ U(i)$.

The states of this chain break into N groups; group I_i has n_i states where n_i is the dimension of A_{ii}. The probabilities of transitioning between states in group I_i are dependent on $A_{ii}(u) + \varepsilon B_{ii}(u)$ and are of high probability relative to transitions between states in different groups I_i and I_j. Transitions from states in I_i to states in I_j are dependent on εB_{ij}. This suggests that transitions within a group take place more frequently than transitions between states in different groups.

For small ε, the behavior within a group can be approximated by the steady state probability density $v_i(u_i)$ of the fast chain

$$v_i(t+1) = v_i(t)\ A_{ii}(u_i), \tag{8}$$

where $u_i \varepsilon R^{ni}$ is the vector of controls corresponding to A_{ii}, after a "relatively" short time. In other words, the probabilities that given states within group I_i will be occupied, given some state in I_i is occupied, is approximately given by $v_i(u_i)$.

Aggregate Markov Chain

The aggregate Markov chain corresponding to the weakly coupled Markov chain is defined by

$$\pi(t+1) = \pi(t)\, Q(u) \tag{9}$$

where

$$Q(u) = I + \varepsilon M(u) \tag{10}$$

and the elements $m_{ij}(u)$ of $M(u)$ satisfy

$$m_{ij}(u_i) = v_i(u_i) B_{ij}(u_i)\, \underline{1} \tag{11}$$

$$\underline{1} = [1\ 1\ \ldots\ 1]' . \tag{12}$$

The number of states, N, in this chain is the number of groups of states I_i in the weakly-coupled Markov chain. The aggregate chain (7) is an approximation for the transition probabilities obtained by aggregating each group of states I_i into a single state. Due to the presence of ε in the transition matrix it provides a slow time-scale model of aggregate changes in p(t).

2. PROPERTIES OF WEAKLY COUPLED MARKOV CHAINS

The properties of weakly coupled Markov chains presented in this section are based on the following assumptions:

(A1) For all $u(i) \varepsilon\ U(i)$, $i=1,\ldots,s$. The weakly coupled Markov chain has a single ergodic class.

(A2) For each group of states I_i and all $u(j) \varepsilon\ U(j)$ $j \varepsilon I_i$ the fast chains (8) each have a single ergodic class.

(A3) For all $u(i) \varepsilon U(i)$, the aggregate chain has a single ergodic class.

For any control

$$u = [u(1) \; u(2) \cdots u(s)]' \tag{13}$$

$$u \varepsilon U = U(1) \times U(2) \times \cdots \times U(s) \tag{14}$$

the steady state probability $p(u)$,

$$p(u) = p(u)[A(u) + \varepsilon B(u)], \tag{15}$$

may be approximated by

$$p(u) = [\pi_1(u)v_1(u_1) \;\big|\; \pi_2(u)v_2(u_2) \;\big|\; \cdots \;\big|\; \pi_N(u)v_N(u_N)] + O(\varepsilon). \tag{16}$$

This result can be proven in various ways [2,3] under A1, A2 and A3. We may interpret (16) as follows: For any initial probability $p(o) = p_o$ and stationary policy u, the probability of being in state $j \varepsilon I_i$ is the probability π_i, of being in aggregate state i multiplied by the probability v_i^j of occupying state j given a state in I_i is occupied.

Note that under A1, g and $A + \varepsilon B$ being continuous functions of u,

$$J(u) = p(u)g(u) \tag{17}$$

where

$$g(u) = [g(1,u(1)) \; g(2,u(2)) \cdots g(s, u(s))]' \tag{18}$$

Substituting (16) into (17) yields

$$J(u) = \sum_{i=1}^{N} \pi_i(u)(v_i(u_i) \cdot g_i(u_i)) + O(\varepsilon) \tag{19}$$

$$\triangleq J_0(u) + O(\varepsilon)$$

where $v_i \cdot g_i$ is a scalar, $g(u)$ is partitioned

$$g(u) = (g_1(u_1) \;\vert\; g_2(u_2) \cdot \cdot \cdot g_N(u_N)) \tag{20}$$

and

$$g_i \varepsilon R^{n_i}, \quad u_i \varepsilon R^{n_i}$$

Denoting

$$\bar{g}_i(u_i) = v_i(u_i) \cdot g_i(u_i) \tag{21}$$

$$\bar{g}(u) = [\bar{g}_1(u_1) \cdot \cdot \cdot \bar{g}_N(u_N)]' \tag{22}$$

we obtain

$$J_0(u) = \pi(u)\, \bar{g}(u) \tag{23}$$

In [3] it is shown that if

$$u_0 = \arg \min_{u \varepsilon U} J_0(u), \quad u^* = \arg \min_{u \varepsilon U} J(u) \tag{24}$$

then

$$\lim_{\varepsilon \to 0} J(u^*) = J_0(u_0). \tag{25}$$

In the next section this result is used to develop a nested optimization of $J_0(u)$ which yields the near optimal control u_0.

3. NESTED OPTIMIZATION

The problem of minimizing $J_0(u)$ subject to the control constraints and the aggregate chain is a slow time-scale optimization; it assumes that each of the fast chains has reached the steady state equilibrium

$v_i(u_i)$. This may in principle be solved by applying Varaiya's algorithm [4,5]. The steps for this problem are as follows:

0) Choose an initial guess for the dual variable, $\bar{C}^0$; set $n=0$

1) For each aggregate state $i \varepsilon \{1,\ldots,N\}$, find

$$u_i^n = \arg\min_{u_i \varepsilon U_i} \left\{ \sum_{j=1}^{N} m_{ij}(u_i)\bar{C}_j^{\,n} + \bar{g}_i(u_i). \right\}. \tag{26}$$

Let $$\bar{h}_i(\bar{C}^n) = \sum_{j=1}^{N} m_{ij}(u_i^{\,n})\bar{C}_j^{\,n} + \bar{g}_i(u_i^{\,n}). \tag{27}$$

Find

$$\bar{h}_{\min}(\bar{C}^n) = \min_{i=1,\ldots,N} \bar{h}_i(\bar{C}^n) \tag{28}$$

$$\bar{h}_{\max}(\bar{C}^n) = \max_{i=1,\ldots,N} \bar{h}_i(\bar{C}^n) \tag{29}$$

$$\hat{h}(\bar{C}^n) = N^{-1} \sum_{j=1}^{N} \bar{h}_i(\bar{C}^n) \tag{30}$$

2) If $h_{\max}(\bar{C}^n) - h_{\min}(\bar{C}^n) < \delta$ stop; set $u_0 = u^n$

3) Compute $\bar{C}_i^{\,n+1} = \bar{C}_i^{\,n} + \Delta[h_i(\bar{C}^n) - \hat{h}(\bar{C}^n)]$ where Δ is a positive stepsize

4) Set $n \leftarrow n+1$; go to 1)

Several points concerning this algorithm should be noted.

- The vector $\bar{C}^n \varepsilon R^N$
- The control u_i for aggregate state i satisfies $u_i \varepsilon R^{n_i}$. This makes the optimization in step 1) difficult at best.

- This slow problem is N dimensional; the aggregate chain transition matrix is NxN.
- The outcome of this optimization is independent of ε.

We next show that the minimization in 1) is an optimization problem for the fast chains.

Factoring $v_i(u_i)$ from equation (26) we obtain

$$u_i{}^n = \arg\min_{u_i \varepsilon U_i} \quad v_i(u_i)\Big(\sum_{j=1}^{N} B_{ij}(u_i)\underline{1}\ \bar{C}_j{}^n + g_i(u_i)\Big) \tag{31}$$

$$\triangleq \arg\min_{u_i \varepsilon U_i} v_i(u_i)\ \tilde{g}(\bar{C}^n,\ u_i)$$

where we have defined a new cost vector $\tilde{g}(\bar{C}^n,\ u_i)$. This optimization is the fast time scale problem to which we now apply Varaiya's algorithm to obtain $u_i{}^n$ for each fast chain.

0) Choose an initial guess $C^0(i,n)$; set k=0. The notation (i,n) denotes the ith fast chain and the nth step in the aggregate chain optimization.

1) For each state j in group I_i (fast chain i) $j=1,\ldots,n_i$, find

$$u^k(j) = \arg\min_{u(j)\varepsilon U(j)} \left\{ \sum_{\substack{\ell=1\\ \ell\neq j}}^{n_i} a_{j\ell}(u(j))C_\ell^k(i,n) + (a_{jj}(u(j))-1)C_\ell^k(i,n) \right.$$

$$\left. + \ \tilde{g}(\bar{C}^n,\ j,\ u(j))\right\} \tag{32}$$

where

$$\tilde{g}(\bar{C}^n,u_i) = [g(\bar{C}^n,1,u(1)) \ \ldots\ \tilde{g}(\bar{C}^n,n_i,u(n_i))]'. \tag{33}$$

Let $$h_j(\bar{C}^n, C^k(i,n)) = \left\{ \sum_{\ell=1}^{N_i} a_{j\ell}(u^k(j))\, C_\ell^k(i,n) + (a_{jj}(u^k(j))-1)C_\ell^k(i,n) + \tilde{g}(\bar{C}^n, j, u^k(j)) \right\} \quad (34)$$

Find

$$h_{\min}(\bar{C}^n, C^k(i,n)) = \min_{j=1,\ldots,n_i} h_j(\bar{C}^n, C^k(i,n)) \quad (35)$$

$$h_{\max}(\bar{C}^n, C^h(i,n)) = \max_{j=1,\ldots,n_i} h_j(\bar{C}^n, C^k(i,n)0 \quad (36)$$

$$\hat{h}(\bar{C}^n, C^k(i,n)) = n_i^{-1} \sum_{j=1}^{n_i} h_j(\bar{C}^n, C^k(i,n)) \quad (37)$$

2) If $h_{\max}(\bar{C}^n, C^k(i,n)) - h_{\min}(\bar{C}^n, C^k(i,n)) < \delta$ stop; set

$$u_i^{\,n} = [u^{(n)}(1) \ldots u^{(n)}(n_i)]' \quad (38)$$

This control is the control to be computed in 1) of aggregate optimization. Set

$$\bar{h}_i(\bar{C}^n) = \hat{h}(\bar{C}^n, C^k(i,n)). \quad (39)$$

3) Compute $C_j^{k+1}(i,n) = C_j^k(i,n) + \Delta[h_j(\bar{C}^n, C^k(i,n)) - h_{\min}(\bar{C}^n, C^k(i,n))]$ where Δ is the stepsize for step k of the fast chain i optimization.

4) Set k←k+1; go to 1).

This algorithm is performed for each fast chain. It replaces equation (26) and (27) in the aggregate optimization. Thus the optimization is nested. For each aggregate state optimization (in a slow time-scale) a fast chain optimization is performed. The nested algorithm has a number of useful properties.

- It is never necessary to compute $\pi(u)$ or $v_i(u_i)$.

- Storing $B_{ij}(u_i)\underline{1}$ rather than $B_{ij}(u_j)$ reduces the number of computations and memory requirements to less than that required for the direct application of Varaiya's algorithm to the original problem.
- The value of small parameter ε does not come into play either in the slow time-scale optimization or the fast time-scale optimization. Thus algorithmic stiffness has been removed and convergence of the nested algorithm should improve.
- When the nested algorithm converges, the control

  $$u^n = [u_1^n | \cdot \cdot \cdot | u_n^n]' . \tag{40}$$

  is u_0 the control minimizing $J_0(u)$.

4. SUMMARY

The result of this paper is a nested optimization algorithm which is designed to find a near-optimal control for weakly coupled Markov chains. The slow time-scale optimization passes a dual vector $\bar{C}$ to N fast time-scale problems. The fast time-scale problems compute a minimizing control and send the control u_i^n and optimal cost $\hat{h}$ back to the aggregate optimization. The result is a decomposition of the problem into N+1 subproblems, N fast time-scale problems and 1 "coordinating" slow time-scale problem.

Weakly coupled Markov chains are one means of formulating hydro-scheduling problems [6,7]. Future work will concentrate on applying this algorithm to that problem.

ACKNOWLEDGEMENT

This paper represents joint work with R. G. Phillips of the Coordinated Sciences Lab at the University of Illinois.

REFERENCES

[1] F. Delebecque and J. P. Quadrat, "Optimal Control of Markov Chains Admitting Strong and Weak Interactions," to appear in Automatica.

[2] V. G. Gaitsgori and A. A. Pervozvanskii, "Aggregation of States in a Markov Chain with Weak Interactions," translation from Kybernetika, No. 3, May - June 1975, pp. 91-98.

[3] D. Teneketzis, S. H. Javid and B. Sridhar, "Control of Weakly Coupled Markov Chains," Proceedings of the 1980 CDC, Albuquerque, N.M.

[4] P. Varaiya, "Optimal and Suboptimal Stationary Controls for Markov Chains," IEEE Trans. on Automat. Contrl., Vol. AC-23, No. 3, June 1978.

[5] J. L. Popyack, R. L. Brown and C. C. White, III, "Discrete Versions of an Algorithm Due to Varaiya," IEEE Trans. on Automat. Control, Vol. AC-24, No. 3, June 1979.

[6] F. Delebecque and J. P. Quadrat, "Contribution of Stochastic Control Singular Perturbation Averaging Team Theories to an Example of Large-Scale Systems: Management of Hydropower Production," IEEE Trans. on Automat. Control, Vol. AC-23, No. 2, April 1978.

[7] D. Teneketzis, S. H. Javid and B. Sridhar, "Operations Scheduling by Markov Chains with Strong and Weak Interactions," Proceedings of the SIAM Conf. on Electric Power Problems A Mathematical Challenges, Seattle, Washington, 1980.

A NEW ALGORITHM FOR NONDIFFERENTIABLE OPTIMIZATION AND ITS USE FOR COMPUTING SADDLE POINTS

Guy COHEN
Centre d'Automatique et Informatique, Ecole Nationale
Supérieure des Mines de Paris, Fontainebleau, France and
INRIA, Le Chesnay, France.

Jean-François BALDUCCHI
Centre d'Automatique et Informatique, ENSMP,
Fontainebleau, France.

ABSTRACT

We first consider the problem of maximizing a concave nondifferentiable functional on a closed convex subset of a Banach space. We propose a new algorithm which yields not only the argument of the maximum in the limit, but also that subgradient involved in the necessary and sufficient optimality conditions. We then show the relevance of this algorithm for computing a saddle point of a convex-concave Lagrangian-like functional when the argument of the inner minimum in the maximin approach is nonunique.

STATEMENT OF THE PROBLEM

To introduce our motivation let us consider the following constrained minimization problem

$$\min_{u \in U} J(u) \quad \text{s.t.} \quad \Theta(u) \in -C \tag{1}$$

where J is a functional on the reflexive Banach space $\mathcal{U}$ ($\mathcal{U}^*$ denotes the topological dual and $\langle .,. \rangle$ the duality product), $U \subset \mathcal{U}$, Θ is a mapping from $\mathcal{U}$ into the Hilbert space $\mathcal{C}$, $C \subset \mathcal{C}$. We recall the following theorem (e.g. see Ekeland and Temam [1]).

Theorem 1. Assume J is l.s.c. and convex, U is closed and convex, C is a closed convex cone with nonempty interior $\overset{\circ}{C}$ whose conjugate cone (see Luenberger [2] is denoted by C^*, assume Θ is C-convex that is

$$\forall u, v \in \mathcal{U},\ \forall \alpha \in [0,1],\ \alpha\Theta(u) + (1-\alpha)\Theta(v) - \Theta(\alpha u+(1-\alpha)v) \in C \tag{2}$$

and such that the functional $u \mapsto \langle p, \Theta(u) \rangle$ is l.s.c. for all $p \in C^*$. Finally, assume that the following constraint qualification condition holds

$$\exists\, \overset{\circ}{u} \in U : \Theta(\overset{\circ}{u}) \in -\overset{\circ}{C} \tag{3}$$

and, if U is not bounded, that $J(u) \to +\infty$ when $\|u\| \to +\infty$ while remaining in U. Then, if we let

$$L(u,p) = J(u) + \langle p, \Theta(u) \rangle, \tag{4}$$

the Lagrangian L has a saddle point on $U \times C^*$. Moreover the set of saddle points (u^*, p^*) is the product of two closed convex sets $U^* \times P^*$ and any $u^* \in U^*$ is a solution of Problem (1) ∎

This dual approach is particularly attractive when one considers a decomposition of $\mathcal{U}$ and U into $\mathcal{U}_1 \times \ldots \times \mathcal{U}_N$ and $U_1 \times \ldots \times U_N$ respectively where U_i is a closed convex subset of the component subspace $\mathcal{U}_i$ and when J and Θ have additive forms, that is

$$J(u) = \sum_{i=1}^{N} J_i(u_i) \quad ; \quad \Theta(u) = \sum_{i=1}^{N} \Theta_i(u_i) \tag{5}$$

(although Cohen [3], [4] extended the following decomposition principle to the general case when these separable forms do not hold). Then L also assumes an additive form with respect to the decomposition in u and this remark is exploited in the Uzawa algorithm, which consists in the following two-level iterations

$$L(u^k,p^k) = \operatorname*{Min}_{u\in U} L(u,p^k) \tag{6}$$

$$p^{k+1} = \Pi(p^k + \rho\Theta(u^k)) \tag{7}$$

where Π denotes the projection onto C^* and ρ is some positive step (see e.g. Cohen [4] for conditions of convergence). As a matter of fact, the lower level minimization (6) splits into N independent minimizations on the U_i's. This is known as the price (Arrow and Hurwicz [5], Lasdon and Schoeffler [6]) or interaction balance (Mesarovic, Macko and Takahara [7]) decomposition-coordination principle.

Another instance when the same kind of decomposition occurs will be given later on. We now want to focuss our attention on the following difficulty. Let

$$\hat{U}(p) = \operatorname{Arg\,Min} L(.,p) = \{u \in U \mid L(u,p) \leq L(v,p),\ \forall\, v \in U\} \tag{8}$$

and consider the case when $\hat{U}(p)$ cannot be garanteed to be a singleton (which implies that $L(u,p)$ is not strictly convex in u). Then the dual functional

$$\psi(p) = \operatorname*{Min}_{u\in U} L(u,p) \tag{9}$$

is nondifferentiable (see Danskin [8], Demyanov and Malozemov [9]). In the present case, ψ is concave in p and subdifferentiable. More precisely, we have the following theorem.

Theorem 2. Under the assumptions of Theorem 1, we have

$$\partial\psi(p) = \overline{\text{co}}\ \Theta(\hat{U}(p)) \tag{10}$$

where $\overline{\text{co}}$ denotes the closed convex hull∎

Note that if $\hat{U}(p)$ is a singleton, then Algorithm (6)-(7) can be interpreted as a projected gradient algorithm for maximizing ψ over C^*. When this is not the case one may think to use (6)-(7) but with ρ replaced by ρ^k at step k, such that

$$\rho^k \geq 0, \quad \sum_{k=0}^{+\infty} \rho^k = +\infty, \quad \sum_{k=0}^{+\infty} (\rho^k)^2 < +\infty. \tag{11}$$

This is a projected subgradient algorithm (see the survey by Polyak [10] for references). This algorithm yields a sequence $\{p^k\}$ whose cluster points (in the weak topology) belong to the optimal set P^*. Unfortunately this does not generally provide a solution to Problem (1) as shown by the following example.

Example 1. Let $\mathcal{U} = \mathcal{C} = \mathbb{R}$, $C = \mathbb{R}^+$, $U = [-10,+10]$, $J(u) = -u$, $\Theta(u) = |u|-1$ in problem (1). We have $U^* \times P^* = \{1\} \times \{1\}$ but $\hat{U}(p^*) = [0,10]$ which contains, but is not reduced to, U^*. Hence, once $p^* = 1$ is known, by

minimizing $L(u,p^*)$, one will not generally obtain the (unique) solution $u^* = 1$ of Problem (1).
This difficulty does not exist if

$$\hat{U}(p^*) = U^* , \quad \forall p^* \in P^*. \tag{12}$$

Iff this property holds, we say, according to Polyak [10], that L is stable in u. However, the only situation when it seems possible to ensure such a property a priori is when L is strictly convex in u. In the other cases, there is almost no numerical algorithm available for computing the u-part of the saddle-point. One exception is the trick proposed in [11] where a strict convexification procedure modifies Problem (6). However this may have drawbacks in some circumstances. For example in the application considered by Balducchi et al. [12], Problem (6) is linear and solved by a very efficient simplex algorithm (Maurras [13]). Therefore, one does not want to strictly convexify it (which implies introducing nonlinearities that prevent using [13]).

In the next section, we show some connections between the problem of finding a saddle point of L and the following wider question. Consider a concave u.s.c. functional ψ on a reflexive Banach space $\mathcal{P}$ and the problem

$$\operatorname*{Max}_{p \in P} \psi(p) \tag{13}$$

where P is a closed convex subset of $\mathcal{P}$. A necessary and sufficient condition for $p^* \in P$ to be a solution of Problem (13) is that

$$\exists\, q^* \in \partial\psi(p^*) \;:\; \langle q^*, p-p^* \rangle \leq 0 , \quad \forall p \in P. \tag{14}$$

The question of interest in the first part of this paper is to compute not only p^*, but a pair (p^*,q^*) meeting (14). We propose a new algorithm that meets this requirement. We then consider applying this algorithm to the saddle point searching problem. In the general case, a further feature must be added to the algorithm in order that it yields a complete answer.

A NEW ALGORITHM FOR NONDIFFERENTIABLE OPTIMIZATION

With the notations already defined, one can characterize a saddle point (u^*,p^*) of L by the following two conditions (plus $p^* \in C^*$)

$$\text{(i)}\;\; u^* \in \hat{U}(p^*) , \qquad \text{(ii)}\;\; \forall\, p \in C^*, \langle p-p^*, \Theta(u^*) \rangle \leq 0. \tag{15}$$

Because of (15-i) and (10), we know that $\Theta(u^*) \in \partial\psi(p^*)$. Therefore (15-ii) should be compared with (14) (while setting $P = C^*$) to show that finding a saddle point of L implies (but is not generally reduced to) finding a pair (p^*,q^*) meeting (14). This is the problem considered in this section.

About these pairs (p^*,q^*) and to strengthen the connection with saddle points, we give the following lemma (that we did not find in the literature) without proof.

Lemma 1. The set of pairs (p^*,q^*) meeting (14) is the product of two convex sets $P^* \times Q^*$ and the functional $(p,q) \mapsto \langle p,q \rangle$ is constant over this set ∎

We consider the following algorithm starting from some $p^o \in P$ and some $q^o \in \partial\psi(p^o)$

$$p^{k+1} = \Pi(p^k + \gamma\, \rho^k q^k) \tag{16}$$

$$q^{k+1} = (1-\rho^{k+1})q^k + \rho^{k+1} r^{k+1} \quad , \quad r^{k+1} \in \partial\psi(p^{k+1}) \tag{17}$$

where Π denotes the projection onto P and γ is a positive constant.
Theorem 3. Let ψ be a concave u.s.c. functional, which is subdifferentiable on P. If the closed convex subset P is not bounded, we further assume that $\psi(p) \to -\infty$ when $\|p\| \to +\infty$ while remaining in P, and that the subgradients of ψ are bounded over P. Then, if $\{\rho^k\}$ is a non-increasing sequence in $]0,1]$ meeting the conditions in (11), the sequence $\{(p^k,q^k)\}$ is bounded and hence has weak cluster points. Any such cluster point is a pair (p^*,q^*) meeting (14). Finally, if ψ is strongly concave (see Auslender [14] for a definition), the sequence $\{p^k\}$ strongly converges towards the unique p^*■

The proof of this theorem cannot be given here because of the lack of space but it will be given in a forthcoming paper together with several ways of avoiding the assumption that the subgradients are bounded and also various ways in which γ in (16) may depend on the step index k.

EXTENSION OF THE ALGORITHM TO SADDLE POINT PROBLEMS

When the functional ψ in (13) is defined by (9) and when $P = C^*$, the assumptions about ψ in Theorem 3 are consequences of the assumptions of Theorem 1 and of the additional assumption that Θ is locally Lipschitz. To implement the above algorithm (16)-(17) in this case, it suffices to interpret r^{k+1} in (17) as $\Theta(u^{k+1})$ where $u^{k+1} \in \hat{U}(p^{k+1})$. This is a consequence of Theorem 2 (see (10)). The cluster points of the sequence $\{(p^k,q^k)\}$ are points (p^*,q^*) such that

$$\text{(i)} \quad q^* \in \overline{\text{co}}\ \Theta(\hat{U}(p^*)), \qquad \text{(ii)} \quad \forall p \in P \ , \quad \langle q^*, p-p^* \rangle \leq 0. \tag{18}$$

But what we would like to have is a pair (u^*,p^*) meeting (15). Hence, in general, a complete answer is not obtained. However there are cases when the problem can essentially be solved by the algorithm in its present form. We describe such a case hereafter.

An example of application.

It is a prototype of problems considered by Cohen [15], a real life example of which has been encountered in [12]. Consider an electrical network with generating facilities that may have stochastic failures and with transmission lines that we want to size in such a way that the sum of investment costs (to build the lines) and of the expectation of the optimal generation costs is minimized. Let $i = 1, \ldots, N$ be the indices of the N stochastic situations corresponding to the various patterns of failures of the generating units, let u_i denote the closed-loop decisions determining the operation of the available generating units and of the network in situation i, let U_i summarize the set of constraints on u_i and the equations of the network, let $J_i(u_i)$ be the generation cost, finally let y_j be the capacity of line j ($j = 1, \ldots, l$) with corresponding cost $c_j y_j$ and let $\Theta_{ji}(u_i)$ be the capacity actually used in situation i.

Then we want to solve

$$\underset{u,y}{\text{Min}} \quad \sum_{j=1}^{l} c_j y_j + \frac{1}{N} \sum_{i=1}^{N} J_i(u_i) \tag{19}$$

$$\text{subject to} \quad y_j \geq 0, \; u_i \in U_i, \quad \Theta_{ji}(u_i) \leq y_j \;\; ; \; i = 1, \ldots, N \;\; ; \; j=1,\ldots,l \tag{20}$$

(for simplicity, we assume that all the situations have the same probability). It can be shown (see Cohen [15]) that this problem is equivalent to finding a saddle point (u^*,p^*) of the Lagrangian

$$L(u,p) = \sum_{i=1}^{N} \left(\frac{1}{N} J_i(u_i) + \sum_{j=1}^{l} c_j p_{ji} \theta_{ji}(u_i) \right) \qquad (21)$$

over the sets $U_1 \times \ldots \times U_N$ and P defined as

$$P = \{p \,|\, p_{ji} \geq 0 \;;\; \sum_{i=1}^{N} p_{ji} = 1 \;;\; \forall j = 1, \ldots, l \;;\; \forall i = 1, \ldots, N\}. \qquad (22)$$

The optimal capacities of the lines are then obtained as

$$y_j^* = \underset{1 \leq i \leq N}{\mathrm{Max}} \; \theta_{ji}(u_i^*) = \sum_{i=1}^{N} p_{ji}^* \; \theta_{ji}(u_i^*) \;;\; j = 1, \ldots, l. \qquad (23)$$

Note that the minimization of L (21) splits into N independent minimizations (and N is very large in this case !). In [12], the J_i's are linear as well as the constraints defining the U_i's, while the θ_{ji}'s are absolute values. Owing to the network equations involved in the definition of the U_i's, one can solve the minimization problems very efficiently by using the Simplex method of Maurras [13]. However the solutions are generally nonunique, especially when p reaches an optimal value p^*.

Keeping all the assumptions of Theorem 1 on J, θ and U, except for assumption (3) which is unnecessary when P is bounded as in (22), one can state the existence of a saddle point of L (21) over $U \times P$ (note that P is included in the positive cone of $\mathbb{R}^m$ with $m = Nl$). Then one can prove the following lemma.

Lemma 2. If a pair (p^*,q^*) meets (18), there exists u^* such that (u^*,p^*) is a saddle point of L over $U \times P$ and

$$\Pi^* q^* = \Pi^* \theta(u^*)$$

where Π^* denotes the projection on the subspace $\{x \in \mathbb{R}^m \,|\, p_i^* = 0 \Rightarrow x_i = 0, \forall i = 1, \ldots, m\}$ ∎

Consequently, the algorithm suggested at the beginning of this section yields $\theta_{ji}(u^*)$ for all pair (j,i) such that $p_{ji}^* > 0$. But these informations are sufficient to compute the optimal capacities of the lines y_j^* according to (23). Moreover if one is interested in reconstructing the whole primal solution u^*, then it suffices to solve N independent problems

$$\underset{u_i \in U_i}{\mathrm{Min}} \; J_i(u_i) \;\text{ s.t. }\; \theta_{ji}(u_i) \leq y_j^* \;;\; j = 1, \ldots, l \;;\; i = 1, \ldots, N$$

which represents a mild extra work.

Extension to general convex problems.

Lemma 2 does not provide the same help as above for solving Problem (1) even when assuming (5). Therefore we propose to add to (16)-(17) (again with $r^{k+1} = \theta(u^{k+1})$ where $u^{k+1} \in \hat{U}(p^{k+1})$) the following parallel iteration

$$v^{k+1} = (1-\rho^{k+1}) v^k + \rho^{k+1} u^{k+1} \qquad (24)$$

starting from $v^0 \in \hat{U}(p^0)$. We proved the following theorem.

Theorem 4. Under the same assumptions as above, the sequence $\{v^k\}$ generated by (24) is bounded and has weak cluster points which belongs to U^* ∎

For the time being we leave as a conjecture that the algorithm

$$p^{k+1} = \Pi(p^k + \gamma\rho^k \Theta(v^k)) \tag{25}$$

together with iteration (24) might have the same property. Note that this is obvious when Θ is affine since both algorithms coincide.

SOME NUMERICAL RESULTS

They have been obtained for a reduced-sized version of problem [12] alluded to in the previous section and that we further detail hereafter. The small network considered here consists of 3 nodes and 3 edges connecting them in a triangle form. There are 10 generating units distributed among the 3 nodes. We considered N=500 randomly selected situations (among the 2^{10} possible ones) for approximating the mathematical expectation of the optimal generation cost (see (19)). In each situation (indexed by i), any generating unit n (among the 10) can deliver a power output of g_{ni} ranging from 0 to $\overline{g}_{ni}$, the latter being either null or equal to the actual maximal power output, depending on whether unit n is in failure or not in situation i. The unit generation cost is denoted by c_n. At each node s(=0,1,2), there is a demand $\overline{d}_s$ to be supplied. Let d_{si} be the part actually supplied in situation i. There is a unit penalty cost e for the part of the demand which is not delivered. Finally, if the line j(=1,2,3) is represented by a directed arc from node j-1 to node j and if z_{ji} represents the algebraic power flow in this line in situation i, the first Kirchoff law (which is the only one retained as the model of transmission in the network) can be written down as

$$z_{si} - z_{s+1,i} + \sum_{n \in N(s)} g_{ni} = d_{si} \quad ; \quad s = 0, 1, 2 \; ; \tag{26}$$

where $z_{0i} \overset{\Delta}{=} z_{3i}$ and N(s) is the set of indices of those generating units which are attached to node s. We have

$$0 \leqslant d_{si} \leqslant \overline{d}_s \quad , \quad 0 \leqslant g_{ni} \leqslant \overline{g}_{ni} \quad ,$$

which describes the admissible set together with (26). The last constraints in (20) should be interpreted as

$$|z_{ji}| \leqslant y_j$$

where y_j is the capacity of transmission line j, with a unit investment cost c_j. Consequently, (19) can now be written as

$$\text{Min} \sum_{j=1}^{3} c_j y_j + \frac{1}{500} \sum_{i=1}^{500} \sum_{s=0}^{2} \left[e(\overline{d}_s - d_{si}) + \sum_{n \in N(s)} c_n g_{ni} \right].$$

Once the p_{ji}'s have been introduced with constraints (22), we consider the Lagrangian (21) and its minimization which splits into 500 (almost) linear programs. Once these have been solved for a given vector p^k, the resulting solutions (indexed by superscript k) are used for updating p^k. Suppose first that we use formula (7) for doing this. The projection Π is now the projection on the set P (22). Since this is not easy to compute, we proposed and justified the following alternative formula in [15]

$$\pi_{ji}^{k+1} = \max\left(0, p_{ji}^k + \gamma\rho^k\left(|z_{ji}^k| - \sum_{i=1}^{500} p_{ji}^k |z_{ji}^k|\right)\right) \tag{27-i}$$

and then $$p_{ji}^{k+1} = \pi_{ji}^{k+1} \Big/ \Big(\sum_{i=1}^{500} \pi_{ji}^{k+1} \Big). \tag{27-ii}$$

However, as dicussed at the beginning of the paper, owing to the absence of strict convexity of the primal problem (an assumption we made in [15]), nonuniqueness of the primal solution may occur (and will almost surely occur as it could be explained heuristically), and this prevents algorithm (27) from yielding the optimal flows z_{ji}^* and consequently the searched optimal capacities

$$y_j^* = \operatorname*{Max}_{1 \leq i \leq 500} |z_{ji}^*| = \sum_{i=1}^{500} p_{ji}^* |z_{ji}^*|. \tag{28}$$

The implementation of algorithm (16)-(17) amounts here to replacing $|z_{ji}^k|$ in (27-i) by q_{ji}^k computed as

$$q_{ji}^k = (1-\rho^k) q_{ji}^{k-1} + \rho^k |z_{ji}^k| \qquad (\text{with } q_{ji}^0 = |z_{ji}^0|). \tag{29}$$

This algorithm has been implemented with the following experimental rule for the adjustment of the weight ρ^k in (29)

$$\rho^k = \frac{A}{1+Bk+(C/\Delta^k)^3} \tag{30}$$

where A, B, C are positive constants and Δ^k represents the discrepancy between the last two terms in (28), but evaluated with q_{ji}^k, namely

$$\Delta^k = \sum_{j=1}^{3} \Big[\operatorname*{Max}_{1 \leq i \leq 500} q_{ji}^k - \sum_{i=1}^{500} p_{ji}^k \, q_{ji}^k \Big].$$

The introduction of the term Δ^k into (30) aims at forcing ρ^k quickly towards zero when we are close to the optimum. Note that, when using the closed-loop formula (30), it is impossible to ensure that conditions (11) will hold if $\Delta^k \to 0$ (but this does not matter in this case) while these conditions can be garanteed if one assume that Δ^k keeps away from zero. But then convergence will be obtained (hence Δ^k will indeed go to zero). As for $\gamma\rho^k$ in (27-i) (where recall that q_{ji}^k should replace $|z_{ji}^k|$), it has been considered globally as if it were a constant all along the iterations. With this adjustment rules for ρ^k and $\gamma\rho^k$ (as a whole), we obtained the convergence rate shown in Fig.1. According to Lemma 2, those q_{ji}^k's for which $p_{ji}^* > 0$ yield the corresponding optimal flows $|z_{ji}^*|$ in the limit, allowing the computation of the optimal capacities through (28). A complete primal solution could be obtained either by solving 500 additional minimization problems after knowing these optimal capacities as explained in the previous section, or by "averaging" the whole primal solution at each step according to (24). Finally our conjecture (25) would amount to using the absolute value of the weighted sum of the z_{ji}^t's, $t \leq k$, in place of $|z_{ji}^k|$ in (27-i) instead of using the weighted sum q_{ji}^k of their absolute values (see (29)).

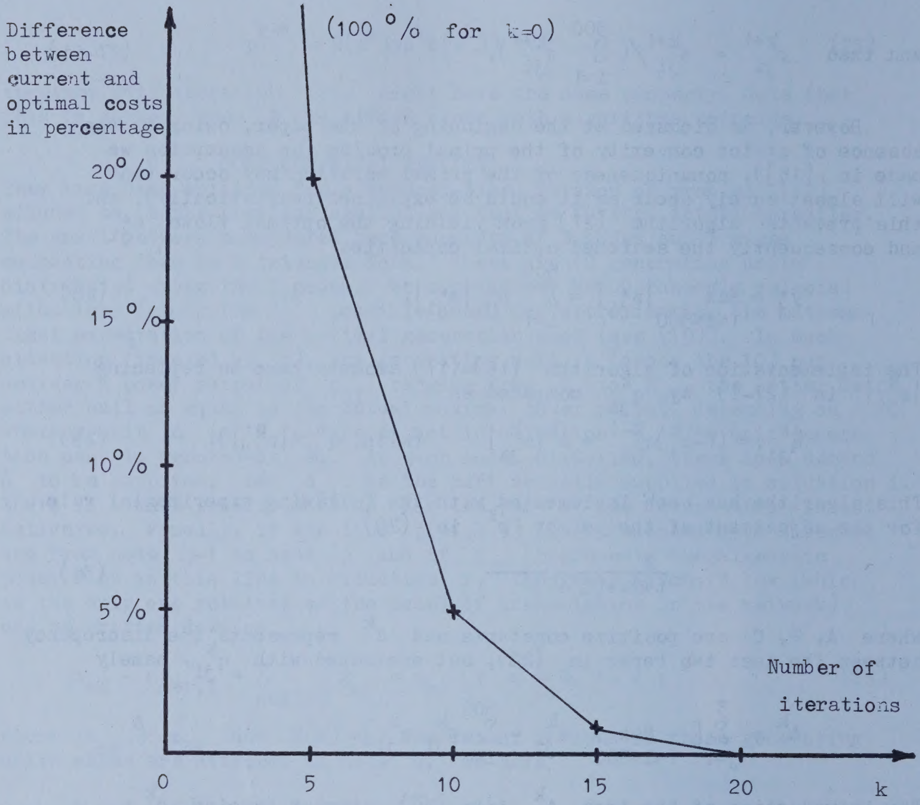


Figure 1. Convergence rate .

CONCLUSION

In this paper, we considered the problem of computing a saddle point of a Lagrangian-like functional in the case when the primal solution is nonunique, and when we do not want to alter the primal minimization problems as proposed in [11]. We established a connection between this problem and that of computing a solution and an "optimal" subgradient (in the sense of (14)) in the framework of a general concave maximization problem (13). We proposed a new algorithm that solves the latter problem and extended it to the saddle point computational problem. We gave some example of application and numerical results. More experience with this algorithm is needed to evaluate its performances in various situations and in comparison with more classical algorithms for nondifferentiable optimization (see [10]). This is under investigation at the present time. Another question, which certainly is crucial as for the efficiency of all these algorithms, is that of practical adjusment rule for the step ρ^k. We here presented an heuristic rule that appeared to be efficient but without theoretical justification.

ACKNOWLEDGMENT

This research has been supported by Electricité De France (EDF).

REFERENCES

[1] I. Ekeland, R. Temam, Convex Analysis and Variational Problems, North Holland, Amsterdam, 1976.

[2] D.G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.

[3] G. Cohen, Optimization by Decomposition and Coordination : a Unified Approach, IEEE Trans. on Automatic Control, Vol. AC-23, No.2, 1978.

[4] G. Cohen, Auxiliary Problem Principle and Decomposition of Optimization Problems, Jal. of Opt. Th. and Appl., Vol. 32, No.3, 1980.

[5] K.J. Arrow, L. Hurwicz, Decentralisation and Computation in Resource Allocation, in Essays in Economics and Econometrics, R.W. Pfouts Ed., Univ. of North Carolina Press, 1960.

[6] L.S. Lasdon, J.D. Schoeffler, A Multilevel Technique for Optimization, Proc. JACC, Troy, New York, 1965.

[7] M.D. Mesarovic, D. Macko, Y. Takahara, Theory of Multilevel Hierarchical Systems, Academic Press, New York, 1970.

[8] V.M. Danskin, The Theory of Max-Min, Springer Verlag, Berlin, 1967.

[9] V.F. Demyanov, V.N. Malozemov, Introduction to Minimax, Wiley, New York, 1974.

[10] B.T. Polyak, Subgradient Methods : a Survey of Soviet Research, in Nonsmooth Optimization, C. Lemaréchal, R. Mifflin Eds., Pergamon, Oxford, 1978.

[11] A.B. Bakushinskii, B.T. Polyak, On the Solution of Variational Inequalities, Soviet Math. Doklady, Vol.15, pp. 1705-1710, 1974.

[12] J.F. Balducchi, G. Cohen, J.C. Dodu, M. Goursat, Hertz, J.P. Quadrat, M. Viot, Three Methods for Optimizing the Capacities of an Electrical Transmission Network, submitted for publication, 1980.

[13] J.F. Maurras, Flot optimal sur un graphe avec multiplicateurs, EDF publication, Etudes et Recherches, Clamart, France, Série S, No.1, 1972.

[14] A. Auslender, Optimisation, Méthodes Numériques, Masson, Paris, 1976.

[15] G. Cohen, An Algorithm for Convex Constrained Minimax Optimization Based on Duality, to appear, 1980.

ROBUST DECENTRALIZED CONTROL USING OUTPUT FEEDBACK*

Ö. Hüseyin**
Univ. of California
Berkeley, CA 94720

M. E. Sezer**
Univ. of Santa Clara
Santa Clara, CA 95053

D. D. Šiljak
Univ. of Santa Clara
Santa Clara, CA 95053

ABSTRACT

A decentralized control scheme is proposed for stabilization of large-scale systems using dynamic controllers with local output feedback. The scheme is robust with respect to uncertainties in parameters and nonlinearities in the interactions among the subsystems.

1. INTRODUCTION

For either conceptual or computational reasons, large scale systems are modeled as interconnections of low-order subsystems. The decomposition approach is a natural modeling scheme when the subsystems can be identified as physical units. It is also very effective in handling the modeling uncertainties by confining them to interconnections among the subsystems. The decentralized control schemes, which are chosen to fit the information structure constraints imposed by the decompositions, can then be made sufficiently robust to tolerate a wide range of nonlinearities and parameter changes in the interconnection structure.

So far, the decentralized control schemes for interconnected systems have been designed using static controllers with local state feedback [1-7]. It is quite common to expect, however, that the entire state of each subsystem is not accessible for control and the control objective has to be achieved using local outputs. Furthermore, the decentralized schemes, which provide local state estimators for the subsystems [5], may be either infeasible (e.g., communication between subsystems is impossible) or uneconomical (e.g., communication between subsystems is too costly) due to the requirement that the local estimators exchange their computed outputs. Under such design constraints, it is reasonable to consider the methods [8, 9] available for decentralized design using dynamic output feedback. These methods, however, are not suitable for stabilization of large-scale systems because they result in closed-loop systems that are not robust, that is, the systems cannot tolerate either modeling errors or nonlinearities unless they are infinitesimally small. Furthermore, the resulting controllers may have the same size as the overall system,thus giving rise to dimensionality problems in the control design.

In this paper, we identify a class of interconnected systems which can always be stabilized by decentralized dynamic output feedback. The proposed control strategy is to stabilize each decoupled subsystem using local dynamic output feedback and, at the same time, make the gains of the loops of the interconnected system sufficiently small so that stability is retained in the presence of interconnections. This strategy, which is based upon the small gain theorem [10], has been used in the context of static state feedback [11] where it produced robust closed-loop systems. The basic configuration of the compensated system in this work consists of a linear, time-invariant system in the forward path, which corresponds to decoupled subsystems, and a memoryless nonlinearity in the feedback path that corresponds

---

* The research reported herein has been supported by the U.S. Department of Energy, Electric Energy Division, under Contract EC-77-S-03-1493.

** On leave from Middle East Technical University, Ankara, Turkey.

to the interconnections. Except for the block-diagonal structure of the forward path system, the configuration is the standard multivariable Lur'e-Postnikov system. In proving stability of the compensated system, we use a small-gain version of the circle criterion [10, 12].

2. PROBLEM STATEMENT

We consider an interconnected system S composed of N linear, time-invariant subsystems S_i described as

$$\begin{aligned} S_i: \dot{x}_i &= A_i x_i + b_i u_i + P_i v_i \\ y_i &= c_i^T x_i \\ w_i &= Q_i x_i , \qquad\qquad i \in N \end{aligned} \tag{2.1}$$

where $x_i(t) \in R^{n_i}$ is the state, $u_i(t) \in R$ is the control input, $y_i(t) \in R$ is the measured output of S_i, and $N = \{1, 2, \ldots , N\}$. The matrices A_i, b_i, P_i, c_i, Q_i are constant and of appropriate dimensions. In (2.1), $v_i(t) \in R^{m_i}$ and $w_i(t) \in R^{\ell_i}$ are the interaction inputs and outputs associated with S_i which represent respectively the effect of other subsystems on S_i, and the effect of S_i on the other subsystems. The interaction inputs and outputs are related as

$$v_i = \phi_i(w) , \quad i \in N \tag{2.2}$$

where the nonlinear function $\phi_i: R^{\ell} \to R^{m_i}$ is continuous in $w = (w_1^T, w_2^T, \ldots, w_N^T)^T$, $\ell = \sum_{i=1}^N \ell_i$, and satisfies the conical condition

$$||\phi_i(w)|| \leq \kappa_i ||w|| , \quad i \in N \tag{2.3}$$

where κ_i is a positive number.

We assume that there exists a subset $M \subset N$ such that

$$P_i = b_i p_i^T , \quad i \in M \tag{2.4a}$$

$$Q_i = q_i c_i^T , \quad i \in N - M . \tag{2.4b}$$

In other words, we assume that for each subsystem S_i either the interaction input v_i has the same effect on S_i as the control input u_i (when $i \in M$), or the interaction output w_i is a reproduction of the measured output y_i (when $i \in N-M$). It should be emphasized that this assumption about the interconnection structure of S is essential in the following development.

We apply to the interconnected system S a decentralized dynamic control

$$\begin{aligned} C_i: \dot{z}_i &= F_i z_i + g_i y_i \\ u_i &= -h_i^T z_i - k_i y_i , \quad i \in N , \end{aligned} \tag{2.5}$$

where $z_i \in R^{r_i}$ is the state of the controller C_i, and F_i, g_i, h_i and k_i are constant and of appropriate dimensions. The compensated subsystem $\hat{S}_i$ consisting of S_i and C_i can be represented as

$$\hat{S}_i: \dot{\hat{x}}_i = \hat{A}_i\hat{x}_i + \hat{P}_i v_i$$
$$w_i = \hat{Q}_i\hat{x}_i , \qquad i \in N \tag{2.6}$$

where $\hat{x}_i = (x_i^T, z_i^T)^T$, and

$$\hat{A}_i = \begin{bmatrix} A_i - b_i k_i c_i^T & -b_i h_i^T \\ g_i c_i^T & F_i \end{bmatrix} , \quad \hat{P}_i = \begin{bmatrix} P_i \\ 0 \end{bmatrix}$$

$$\hat{Q}_i = [Q_i \quad 0] \qquad i \in N . \tag{2.7}$$

Our goal is to choose the order r_i and the parameters F_i, g_i, h_i and k_i of the controllers C_i such that the null solution $\hat{x}_i = 0$, $i \in N$, of the compensated interconnected system $\hat{S}$ consisting of the compensated subsystems $\hat{S}_i$ of (2.6) and the interconnections in (2.2) is asymptotically stable in the large for all nonlinearities ϕ_i satisfying (2.3), that is, $\hat{S}$ is absolutely stable. For this purpose, we first investigate the effect of proper choice of each individual controller C_i on the subsystem $\hat{S}_i$.

3. SUBSYSTEM ANALYSIS

In order to simplify the notation in this section, we drop the subscript i from the subsystem S_i of (2.1) so that

$$S_i: \dot{x} = Ax + bu + pv$$
$$y = c^T x$$
$$w = q^T x \quad . \tag{3.1}$$

where $x \in R^n$. Furthermore, we assume that the interaction input v and interaction output w of S_i are scalars, thus u, v, y, w $\in R$. We assume also that S_i is controllable by the control input u, and observable from the measured output y, i.e., the triple (A,b,c^T) is controllable and observable.

Similarly, we represent a typical controller C_i of (2.5) associated with S_i as

$$C_i: \dot{z} = Fz + gy$$
$$u = -h^T z - ky , \tag{3.2}$$

where $z \in R^r$. The compensated subsystem $\hat{S}_i$ consisting of S_i in (3.1) and C_i in (3.2) can be described by

$$\hat{S}_i: \dot{\hat{x}} = \hat{A}\hat{x} + \hat{p}v$$
$$w = \hat{q}^T\hat{x} , \tag{3.3}$$

where $\hat{x} = (x^T, z^T)^T$, and

$$\hat{A} = \begin{bmatrix} A - bkc^T & -bh^T \\ gc^T & F \end{bmatrix} , \quad \hat{p} = \begin{bmatrix} p \\ 0 \end{bmatrix}$$

$$\hat{q}^T = [q^T \quad 0]. \tag{3.4}$$

We now define the following transfer functions:

$$\begin{aligned} G_{yu}(s) &= c^T(sI-A)^{-1}b\ , \\ G_{yv}(s) &= c^T(sI-A)^{-1}p\ , \\ G_{wu}(s) &= q^T(sI-A)^{-1}b\ , \\ H(s) &= k + h^T(sI-F)^{-1}g \\ W(s) &= \hat{q}^T(sI-\hat{A})^{-1}\hat{p}\ . \end{aligned} \tag{3.5}$$

If $p = b$ in (3.1), then by direct computation we obtain

$$W(s) = \frac{G_{wu}(s)}{1 + G_{yu}(s)H(s)} \tag{3.6}$$

Similarly, if $q = c$ in (3.1), then $W(s)$ can be computed as

$$W(s) = \frac{G_{yv}(s)}{1 + G_{yu}(s)H(s)} \tag{3.7}$$

Since the transfer functions in (3.6) and (3.7) are of the same form (in fact, they have the same poles), the following argument applies to both of them equally well. Therefore, in the remaining of this section we shall assume that $p = b$ in (3.1), so that $W(s)$ is as given in (3.6).

We begin our development by writing $G_{yu}(s)$ and $G_{wu}(s)$ in (3.5) explicitly as

$$\begin{aligned} G_{yu}(s) &= \beta_0 \frac{\beta(s)}{\alpha(s)} \\ &= \beta_0 \frac{s^{n-\nu-1} + \beta_1 s^{n-\nu-2} + \dots + \beta_{n-\nu-1}}{s^n + \alpha_1 s^{n-1} + \dots + \alpha_n} \end{aligned} \tag{3.8}$$

where $0 \le \nu \le n-1$, and

$$\begin{aligned} G_{wu}(s) &= \frac{\gamma(s)}{\alpha(s)} \\ &= \frac{\gamma_1 s^{n-1} + \gamma_2 s^{n-2} + \dots + \gamma_n}{s^n + \alpha_1 s^{n-1} + \dots + \alpha_n}\ . \end{aligned} \tag{3.9}$$

In writing (3.8) we assume that $G_{yu}(s)$ has $n-\nu-1$ zeros. We shall further assume that these zeros lie in the open left-half complex plane P^-. We choose the order of C to be ν, i.e., $r = \nu$. Then, $H(s)$ in (3.5) has the form

$$\begin{aligned} H(s) &= \eta_0 \frac{\eta(s)}{\sigma(s)} \\ &= \eta_0 \frac{s^\nu + \eta_1 s^{\nu-1} + \dots + \eta_\nu}{s^\nu + \sigma_1 s^{\nu-1} + \dots + \sigma_\nu}\ . \end{aligned} \tag{3.10}$$

Finally, using (3.8)-(3.10) we can express $W(s)$ in (3.6) as

$$W(s) = \frac{\gamma(s)\sigma(s)}{\alpha(s)\sigma(s) + \beta_0\eta_0\beta(s)\eta(s)}\ . \tag{3.11}$$

We now choose the parameters of the controller C as follows:

(i) η_j, $j = 1, 2, \dots \nu$, , in (3.10) are chosen arbitrarily subject to the condition that $\eta(s)$ has all its zeros in P^-.

(ii) The coefficients σ_j, $j = 1, 2, \dots \nu$, , and η_0 in (3.10) are chosen as

$$\sigma_j = \tilde{\sigma}_j \rho^j, \quad j = 1, 2, \dots, \nu,$$

$$\beta_0 \eta_0 = \tilde{\sigma}_{\nu+1} \rho^{\nu+1}, \tag{3.12}$$

where $\rho > 0$ is a parameter to be specified, and $\tilde{\sigma}_j$, $j = 1, 2, \dots, \nu+1$ are such that the polynomial

$$\tilde{\sigma}(s) = s^{\nu+1} + \tilde{\sigma}_1 s^{\nu} + \tilde{\sigma}_2 s^{\nu-1} + \dots + \tilde{\sigma}_{\nu+1} \tag{3.13}$$

has distinct zeros all in P^-.

Denoting by P^+ the closed right-half complex plane we now state the following lemma, whose proof is given in the appendix.

(3.14) LEMMA. *Under the conditions* (i) *and* (ii) *above, for any* $\varepsilon > 0$ *there exists a* $\bar{\rho} > 0$ *such that whenever* $\rho > \bar{\rho}$ *the following conditions are satisfied:*

(i) $\hat{A}$ has all eigenvalues in P^-.

(ii) $|W(s)| < \varepsilon$ for all $s \in P^+$.

The properties of the closed-loop system transfer function $W(s)$ stated in the above lemma enable us to develop a stabilization procedure for the interconnected system S described in (2.1). This we consider in the next section.

4. MAIN RESULT

The preliminaries in the previous section have already given the clue about the mechanism of achieving stability of the closed-loop interconnected system: a small gain version of the circle criterion [10, 12], which we summarize below.

Let us consider the compensated interconnected system $\hat{S}$, which can be described in compact form as

$$\hat{S}: \dot{\hat{x}} = \hat{A}\hat{x} + \hat{P}\Phi(w)$$
$$w = \hat{Q}\hat{x}, \tag{4.1}$$

where $\hat{x} = (\hat{x}_1^T, \hat{x}_2^T, \dots, \hat{x}_N^T)^T$, $\Phi(w) = [\phi_1^T(w), \phi_2^T(w), \dots, \phi_N^T(w)]^T$, and $\hat{A} = \text{diag}\,(\hat{A}_1, \hat{A}_2, \dots, \hat{A}_N)$, $\hat{P} = \text{diag}\,(\hat{P}_1, \hat{P}_2, \dots, \hat{P}_N)$, $\hat{Q} = \text{diag}\,(\hat{Q}_1, \hat{Q}_2, \dots, \hat{Q}_N)$, with $\hat{P}_i = [P_i^T \;\; 0^T]^T$, $\hat{Q}_i = [Q_i \;\; 0]$. The linear part of $\hat{S}$ has the transfer function

$$W(s) = \hat{Q}(sI-\hat{A})^{-1}\hat{P}, \tag{4.2}$$

which is in block diagonal form, that is, $W(s) = \text{diag}\,[W_1(s), W_2(s), \dots, W_N(s)]$, with the diagonal blocks

$$W_i(s) = \hat{Q}_i(sI-\hat{A}_i)^{-1}\hat{P}_i, \quad i \in N \tag{4.3}$$

corresponding to the compensated subsystems $\hat{S}_i$ of (2.6). Also, it follows from (2.3) that the aggregate nonlinearity Φ: $R^{\ell} \to R^m$, $m = \sum_{i=1}^N m_i$, satisfies

$$||\Phi(w)|| \leq \kappa \, ||w||, \tag{4.4}$$

for some positive number κ.

We now use the following result of reference [12] about the absolute stability of $\hat{S}$ described by (4.1):

(4.5) LEMMA. *Suppose the matrix* $\hat{A}$ *has all its eigenvalues in* P^-. *Then, the system* $\hat{S}$ *in* (4.1) *is absolutely stable if*

$$\sup_{\omega \in R} \max_k |\lambda_k[W^T(-j\omega)W(j\omega)]| < \frac{1}{\kappa^2}, \tag{4.6}$$

where λ_k *denotes the eigenvalues of the indicated matrix.*

Using Lemmas (3.14) and (4.5), we arrive at our main result;

(4.7) THEOREM. *Suppose that the triplets* (A_i, b_i, c_i^T) *of* (2.1) *are controllable and observable and, in addition, have all the zeros in* P^-. *Suppose also that the pairs* (P_i, Q_i) *satisfy the condition* (2.4). *Then, there exist a set of local controllers* C_i, $i \in N$, *of* (2.5) *such that the compensated interconnected system* $\hat{S}$ *of* (4.1) *is absolutely stable.*

PROOF. By the condition (2.4), the subsystem transfer functions $W_i(s)$ of (4.3) have the forms

$$W_i(s) = \hat{Q}_i(sI-\hat{A}_i)^{-1}\hat{b}_i p_i^T, \quad i \in M \tag{4.8a}$$

$$W_i(s) = q_i\hat{c}_i^T(sI-\hat{A}_i)^{-1}\hat{P}_i, \quad i \in N-M \tag{4.8b}$$

where $\hat{A}_i$ corresponds to $\hat{A}$ in (3.4), and $\hat{b}_i$ and $\hat{c}_i^T$ are obtained from b_i and c_i by augmenting with zeros. Then, it follows from Lemma (3.14) that under the conditions of the theorem the local controllers C_i can be designed such that the matrices A_i have all their eigenvalues in P^-, and that for any set of ε_i, $i \in N$,

$$||W_i(s)|| < \varepsilon_i, \tag{4.9}$$

for all $s \in P^+$. In particular, (4.9) holds for all $s = j\omega$, $\omega \in R$. Thus, the matrix $\hat{A}$ of (4.1) has all its eigenvalues in P^-, and letting $\varepsilon_i = 1/\kappa^2$, the transfer function $W(s)$ in (4.2) satisfies (4.6). Then, the proof follows from Lemma (4.5).

We note that the conditions (2.4) have a similar structural interpretation as those derived in the context of decentralized stabilization by static state feedback [11, 13]. Constraining the structure of interaction among the subsystems, the conditions (2.4) make it possible to design local controllers which can stabilize the interconnected system independent of the size of the interconnections so long as they are bounded. The conditions (2.4), however, are not the weakest conditions for decentralized stabilizability, and one can expect to obtain less restrictive conditions by a more detailed analysis of the effect of local controllers on the subsystems and on the interconnections, as has been done in the case of state feedback [11]. We should also note that the conditions (2.4) can be a promising starting point to attack the problem of suboptimal design of local controllers, the state feedback version of which has been considered in [13].

REFERENCES

[1] E. J. Davison: The decentralized stabilization and control of a class of unknown nonlinear time-varying systems. Automatica, vol. 10, pp. 309-316, 1974.

[2] M. Ikeda, O. Umefuji, and S. Kodama: Stabilization of large-scale linear systems. Systems, Computers, and Control, vol. 7, pp. 34-41, 1976.

[3] D. D. Šiljak and M. B. Vukčević: Decentrally stabilizable linear and bilinear systems. International Journal of Control, vol. 26, pp. 289-305, 1977.

[4] M. E. Sezer and Ö. Hüseyin: Stabilization of linear time-invariant interconnected systems using local state feedback. IEEE Transactions, vol. SMC-8, pp. 751-756, 1978.

[5] D. D. Šiljak: Large-Scale Dynamic Systems: Stability and Structure, North-Holland, New York, 1978.

[6] M. E. Sezer and Ö. Hüseyin: On decentralized stabilization of interconnected systems, Automatica, vol. 16, pp. 205-209, 1980.

[7] M. Ikeda and D. D. Šiljak: On decentrally stabilizable large-scale systems. Automatica, vol. 16, pp. 331-334, 1980.

[8] S. H. Wang and E. J. Davison: On the stabilization of multivariable decentralized systems. IEEE Transactions, vol. AC-18, pp. 473-478, 1973.

[9] J. P. Corfmat and A. S. Morse: Decentralized control of linear multivariable systems. Automatica, vol. 12, pp. 479-495, 1976.

[10] G. Zames: On the input-output stability of time-varying nonlinear feedback systems. Part I: Conditions derived using concepts of loop gain, conicity, and positivity. IEEE Transactions, vol. AC-11, pp. 228-238, 1966.

[11] M. E. Sezer and D. D. Šiljak: Stability of large-scale systems using graph-theoretic concepts. Proceedings of the Joint Automatic Control Conference, Paper No. WA1-D, 1980.

[12] P. Molander: Stabilization of Uncertain Systems. Department of Automatic Control, Lund Institute of Technology, Lund, Sweden, Report, no. CODEN: LUFTD2/TFRT-1020)/1-111, 1979.

[13] M. E. Sezer and D. D. Šiljak: Robustness of Suboptimal Control: Gain and Phase Margin. Proceedings of the 19th IEEE Conference on Decision and Control, Albuquerque, New Mexico, Paper No. FA4-10:15, 1980.

[14] K. Inan: Asymptotic Root Loci in Linear Multivariable Control Systems. Dissertation, Middle East Technical University, Ankara, Turkey, 1978.

APPENDIX: PROOF OF LEMMA (3.14)

Let W(s) in (3.11) be written as

$$W(s) = \frac{\psi(s)}{\theta(s)}$$

$$= \frac{\psi_1 s^{n+\nu-1} + \psi_2 s^{n+\nu-2} + \dots + \psi_{N+\nu}}{s^{n+\nu} + \theta_1 s^{n+\nu-1} + \dots + \theta_{n+\nu}}, \tag{A.1}$$

where ψ_j and θ_j, $j = 1, 2, \dots, n+\nu$, are functions of the parameter ρ. Comparing (A.1) and (3.11), and using (3.12) we have

$$\theta(s) = s^{\nu}\alpha(s) + \rho\tilde{\sigma}_1 s^{\nu-1}\alpha(s) + \dots + \rho^{\nu}\tilde{\sigma}_{\nu}\alpha(s) + \rho^{\nu+1}\tilde{\sigma}_{\nu+1}\beta(s)\eta(s). \tag{A.2}$$

The proof of Lemma (3.14) is based on the following variation of a result by Inan [14]:

(A.3) LEMMA. *As* $\rho\to\infty$, k+1 *zeros of* $\theta(s)$ *approach to* ρ *times the zeros of* $\tilde{\sigma}(s)$ *in* (3.13), *while the remaining* n-1 *zeros of* $\theta(s)$ *approach to the zeros of* $\beta(s)\eta(s)$ *in the following sense:*

(i) *Let* μ *be a zero of* $\tilde{\sigma}(s)$ *(which is, by assumption, simple). Then there exist* $M_1 > 0$ *and* $\bar{\rho}_1 > 0$ *such that an exact zero* $\mu(\rho)$ *of* $\theta(s)$ *satisfy*

$$|\mu(\rho) - \rho\mu| \leq M_1 \text{ for all } \rho > \bar{\rho}_1 .$$

(ii) *Let* λ *be a zero of* $\beta(s)\eta(s)$ *with multiplicity* ℓ. *Then, there exist* $M_2 > 0$ *and* $\bar{\rho}_2 > 0$ *such that exactly* ℓ *zeros,* $\lambda_j(\rho)$, $j = 1, 2, \dots, \ell$, *of* $\theta(s)$ *satisfy*

$$|\lambda_j(\rho) - \lambda| \leq M_2\rho^{-1/\ell} \text{ for all } \rho > \bar{\rho}_2 .$$

Now, consider $\theta(s)$ in (A.2), where $\beta(s)$ has its zeros in P^- by assumption, and $\eta(s)$ and $\tilde{\sigma}(s)$ have their zeros in P^- by the choice of the parameters of the controller C. Then Lemma A.3 ensures that there exists $\bar{\rho}_3 > 0$ such that $\theta(s)$ has all its zeros in P^- for all $\rho > \bar{\rho}_3$. Since the triple (A,b,c^T) is controllable and observable, the proof of the first part of Lemma (3.14) is complete. To prove the second part let us denote the zeros of $\tilde{\sigma}(s)$ by μ_j, $j = 1, 2, \dots, \nu+1$, and those of $\beta(s)\eta(s)$ by λ_j, $j = 1, 2, \dots, n-1$. Also let $\mu_j(\rho)$, $j = 1, 2, \dots, \nu+1$, and $\lambda_j(\rho)$, $j = 1, 2, \dots, n-1$, denote the exact zeros of $\theta(s)$ which approach to $\rho\mu_j$ and λ_j respectively as $\rho \to \infty$. Let us rewrite $W(s)$ in (A.1) as

$$\begin{aligned} W(s) &= \frac{\psi(s)}{\prod_{j=1}^{\nu+1} [s-\mu_j(\rho)] \prod_{j=1}^{n-1} [s-\lambda_j(\rho)]} \\ &= \frac{\psi(s)}{\prod_{j=1}^{\nu+1} (s-\rho\mu_j) \prod_{j=1}^{n-1} (s-\lambda_j)} \cdot \prod_{j=1}^{\nu+1} \frac{s-\rho\mu_j}{s-\mu_j(\rho)} \cdot \prod_{j=1}^{n-1} \frac{s-\lambda_j}{s-\lambda_j(\rho)} \\ &= \frac{\sigma(s)}{\prod_{j=1}^{\nu+1} (s-\rho\mu_j)} \cdot \frac{\gamma(s)}{\prod_{j=1}^{n-1} (s-\lambda_j)} \cdot \prod_{j=1}^{\nu+1} \frac{s-\rho\mu_j}{s-\mu_j(\rho)} \cdot \prod_{j=1}^{n-1} \frac{s-\lambda_j}{s-\lambda_j(\rho)} . \end{aligned} \tag{A.4}$$

Since $\gamma(s)$ has degree $n-1$, and $\lambda_j \in P^-$, $j = 1, 2, \dots, n-1$, we have for some $M_3 > 0$

$$\left| \frac{\gamma(s)}{\prod_{j=1}^{n-1} (s-\lambda_j)} \right| < M_3 \quad \text{for all} \quad s \in P^+ . \tag{A.5}$$

On the other hand, using the result of Lemma A.3 it is easy to show that there exists $\bar{\rho}_4 > 0$ such that for $\rho > \bar{\rho}_4$

$$\left| \frac{s-\rho\mu_j}{s-\mu_j(\rho)} \right| < 2 \quad \text{for all} \quad s \in P^+ , \; j = 1, 2, \dots, \nu+1, \tag{A.6a}$$

and

$$\left| \frac{s-\lambda_j}{s-\lambda_j(\rho)} \right| < 2 \quad \text{for all} \quad s \in P^+ , \; j = 1, 2, \dots, n-1 . \tag{A.6b}$$

Now, consider the first term on the right-hand side of (A.4), which can be expanded into partial fractions as

$$\frac{\sigma(s)}{\prod_{j=1}^{\nu+1}(s-\rho\mu_j)} = \sum_{j=1}^{\nu+1} \frac{R_j(\rho)}{s-\rho\mu_j}\,, \tag{A.7}$$

where

$$R_j(\rho) = \frac{\sigma(\rho\mu_j)}{\prod_{\substack{\ell=1\\ \ell\neq j}}^{\nu+1}(\rho\mu_j-\rho\mu_\ell)}\,, \quad j = 1, 2, \ldots, \nu+1\,. \tag{A.8}$$

Using (3.12), $R_j(\rho)$ in (A.8) can be computed as

$$R_j(\rho) = \frac{(\rho\mu_j)^\nu + \tilde{\sigma}_1\rho(\rho\mu_j)^{\nu-1} + \ldots + \tilde{\sigma}_\nu\rho^\nu}{\prod_{\substack{\ell=1\\ \ell\neq j}}^{\nu+1}(\rho\mu_j-\rho\mu_\ell)}$$

$$= \frac{\mu_j^\nu + \tilde{\sigma}_i\mu_j^{\nu-1} + \ldots + \tilde{\sigma}_\nu}{\prod_{\substack{\ell=1\\ \ell\neq j}}^{\nu+1}(\mu_j-\mu_\ell)}\,, \tag{A.9}$$

so that

$$|R_j(\rho)| = M_{4j} < \infty\,, \quad j = 1, 2, \ldots, \nu+1\,. \tag{A.10}$$

Since $\mu_j \in P^-$, (A.10) implies that

$$\left|\frac{R_j(\rho)}{s-\rho\mu_j}\right| \leq \frac{M_{4j}}{\rho|\mathrm{Re}\ \mu_j|}\,, \quad j = 1, 2, \ldots, +1\,,$$

for all $s \in P^+$, so that from (A.7)

$$\left|\frac{\sigma(s)}{\prod_{j=1}^{\nu+1}(s-\rho\mu)}\right| \leq \frac{1}{\rho}\sum_{j=1}^{\nu+1}\frac{M_{4j}}{|\mathrm{Re}\ \mu_j|} = \frac{M_4}{\rho}\ . \tag{A.11}$$

From (A.4)-(A.6) and (A.11) it follows that, if $\rho > \max\{\bar{\rho}_3, \bar{\rho}_4\}$, then

$$|W(s)| < \frac{1}{\rho}\, 2^{n+\nu}\, M_3M_4\,, \tag{A.12}$$

for all $s \in P^+$. Now, for any $\varepsilon > 0$, letting $\bar{\rho}_5 = \dfrac{2^{n+\nu}M_3M_4}{\varepsilon}$ and $\bar{\rho} = \max\{\bar{\rho}_3, \bar{\rho}_4, \bar{\rho}_5\}$, it follows that

$$|W(s)| < \varepsilon \tag{A.13}$$

for all $\rho > \bar{\rho}$ and for all $s \in P^+$ completing the proof of the lemma.

EIGENVALUE ASSIGNMENT VIA DECENTRALIZED STATE FEEDBACK

S. LEFEBVRE, S. RICHTER and R. DECARLO
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

A continuations approach is used to construct a new algorithm for designing local state feedback controllers which assign prescribed set of eigenvalues. The large scale system model used in this work is the Component Connection Model. In this model, components and connections have separate defining equations. The continuations algorithm results from an appropriate parametrization of the connections.

INTRODUCTION

Two approaches seem to dominate research in the area of large scale systems. The first approach uses a classical composite system state model upon which researchers impose physically motivated but very often artificial partitioning. The second perspective views a large scale system as an interconnection of subsystems or components. Consistent with this philosophy, is the idea of a decoupled model. The Component Connection Model, CCM, is such a model. Two separate but coupled sets of equations comprise the CCM. One set models the dynamics of each component or subsystem irrespective of their interconnections. The second set models the interconnections which are conservation laws such as KVL and KCL. This model is more restrictive in that it is an interconnection of state models. However, it is highly non-trivial and can adequately characterize many large scale systems.

Most of the research effort in decentralized control has concentrated on the classical composite system state models. The first contribution to the problem of eigenvalue assignment or stabilization by decentralized control was the introduction by Wang and Davison [1] of the fixed mode concept and a theorem showing that the fixed modes of a system cannot be assigned by local dynamic compensation. Later in 1976, Corfmat and Morse [2] reexamined the problem and gave a more complete characterization of the conditions for spectral assignment in terms of systems parameters and properties. Recently Saeks showed that the fixed modes of the composite system and the set of fixed modes of the individual components coincide in the Component Connection Model context [3].

The above papers imply that in the CCM framework there exists local dynamic compensation which assigns a given composite spectrum provided there exists local constant feedback which moves all the system eigenvalues. Neither paper however suggests how to design these local compensators. The present paper addresses this problem in the CCM setting. A recursive algorithm based on a continuations approach is shown to provide the desired local state feedback matrices. An example is given.

COMPONENT CONNECTION METHOD

The decoupled model intrinsic to the approach taken in this paper is the Component Connection Model (CCM) [4]. The CCM comprises two separate but coupled equations. One set models component (subsystem) dynamics which are assumed linear or linearized. The other set models the connections which must be linear and algebraic. The CCM is illustrated in Figure 1. Let the i-th component have input vector a_i, output vector b_i, and state vector x_i. The linear state model for this component takes the well-known form:

$$\begin{aligned} \dot{x}_i &= A_i x_i + B_i a_i \\ b_i &= C_i x_i + D_i a_i \end{aligned} \tag{1}$$

Suppose there are N components. To form the composite component model, stack the N component equations as

$$\begin{aligned} \dot{x} &= Ax + Ba \\ b &= Cx + Da \end{aligned} \tag{2}$$

where $x = \begin{bmatrix} x_1, \ldots, x_N \end{bmatrix}^T$, $a = \begin{bmatrix} a_1, \ldots, a_N \end{bmatrix}^T$, $b = \begin{bmatrix} b_1, \ldots, b_N \end{bmatrix}^T$, $A = \text{block-diag}\{A_1, \ldots, A_N\}$, etc. The total number of states n is equal to the sum of the number of states of each component, $n = \sum_{i=1}^{N} n_i$. Note that the number of components N is generally not equal to the number of states. Denoting u as the system input vector, y as the system output vector, and L_{ij} as appropriately dimensioned connection matrices, then the component and system interconnections take the form

$$\begin{aligned} a &= L_{11} b + L_{12} u \\ y &= L_{21} b + L_{22} u \end{aligned} \tag{3}$$

The connection matrices are typically sparse.

From a decentralized control perspective, system inputs and system outputs are not the significant variables. Rather the component inputs and component outputs play the important role. Consequently in this development the component connection model will consist of equation (2) and the truncated connection equation:

$$a = L_{11} b \triangleq Lb \tag{4}$$

Note in (4) the subscripts on L_{11} are dropped. The reasoning is that in the decentralized context the state of the i-th component will drive only the input of the i-th component either through a constant feedback matrix or through a local dynamic compensator built into the component equations.

If the composite component state vector x coincides with the composite system state vector, then the CCM (as defined by equations ((2), (4)) gives rise to a composite state model [4] of the form

$$\begin{aligned} \dot{x} &= Fx + Ba \\ b &= Cx + Da \end{aligned} \tag{5}$$

where B, C and D are as before and F is given in (6). The matrix

$(I - DL)^{-1}$ exists generically [4].

$$F = A + BL(I - DL)^{-1}C \qquad (6)$$

The decentralized control structure is illustrated in Figures 2 and 3. Define $a_i \leftarrow a_i + K_{di}x_i$. The problem is to find K_d = block-diag $[K_{D1}, \ldots, K_{dN}]$ such that a prescribed composite spectrum is assigned.

CONTINUATIONS APPROACH

The idea behind the computational algorithm is the following. Suppose the spectrum Λ must be assigned to the composite system by decentralized feedback (i.e., the system described by equations (5-6)). Then this spectrum is first assigned to the composite component model which is given in equation (2). This is easy since the components are decoupled. Then the connections are continuously introduced and the local controllers are continuously modified so that the eigenvalues remain fixed in presence of stronger interaction. The basic equations of this continuations approach are derived below.

Consider parametrizing the connections in equaton (4) as rL with $0 \leq r \leq 1$. Then denote

$$F(r) = A + B(rL)\ (I-D(rL))^{-1}C \qquad (7)$$

Clearly, at r = 0 the system is decoupled and at r = 1 it is fully connected. Let σ stand for the spectrum of a matrix. Let $K_d(r)$ be the feedback matrix for the particular parametrization such that

$$\sigma(F(r) + BK_d(r)) = \Lambda \qquad (8)$$

where Λ is the specified spectrum (which is fixed). At r = 0 the matrix $K_d(0)$, which assigns the specified eigenvalues, is easily computed because the system is decoupled. The idea, as introduced above, is to slowly introduce the connections using discrete increments Δr and to modify $K_d(r)$ so that the eigenvalues remain unchanged in presence of the stronger interactions. The advantage of this iterative approach is that updating $K_d(r)$ to obtain $K_d(r+\Delta r)$ is much easier than trying to directly compute $K_d(1)$, the final answer. The way in which the matrix $K_d(r+\Delta r)$ is obtained requires some explanation.

The j-th eigenvalue equation for any value of r may be written as

$$[\lambda_j(r) - A - rBL(I-rDL)^{-1}\ C - BK_d(r)\ ]\ e_j(r) = 0 \qquad (9)$$

In equation (9) $e_j(r)$ is known as a right eigenvector. The system also has a left j-th eigenvector $\underline{e}_j$ with the property

$$\underline{e}_j^H\ [\lambda_j(r) - A - rBL(I-rDL)^{-1}\ C - BK_d(r)\ ] = 0 \qquad (10)$$

Note the superscript H stands for complex conjugate transposed. Differentiating (9) with respect to r, multiplying on the left by $\underline{e}_j^H$, and rearranging while suppressing specific r-dependencies produces

$$\underline{e}_j^H B \frac{dK_d}{dr} e_j = - \underline{e}_j^H BL(I-rDL)^{-2} Ce_j \tag{11}$$

The superscript -2 stands for inverse squared. Note that in (11) the only unknown is $\frac{dK_d}{dr}$. Solution of (11) for j = 1 to n yields $\frac{dK_d}{dr}$ and permits the desired update in the form

$$K_d(r+\Delta r) \simeq K_d(r) + \Delta r \frac{dK_d}{dr} \Big|_r \tag{12}$$

Solving for $\frac{dK_d}{dr}$ is much easier than it may seem at first sight. This task is facilitated by using the tensor (Kronecker) product operation. Suppose $M = \{m_{ij}\}$ is a p x q matrix and N is r x s, then the tensor product of M by N, $M \otimes N$, is defined to be the pr x qs matrix $M \otimes N = \{m_{ij} N\}$. The "vec" operation on the matrix $M = [M_1 | M_2 | \ldots | M_N]$, where M_i is a column of M, is also useful. By definition

$$\text{vec}\ (M) \triangleq \begin{bmatrix} M_1 \\ M_2 \\ \cdot \\ \cdot \\ M_N \end{bmatrix} \tag{13}$$

that is, vec(M) lists the entries of M column-wise. It can be shown that the set of equations (11) for j=1 to n is equivalent to

$$\hat{P}(r)\ \text{vec}(\frac{dK_d}{dr}(r)) = \beta(r) \tag{14}$$

where the matrix $\hat{P}(r)$ is

$$\hat{P}(r) = \begin{bmatrix} (e_1)^T \otimes (\underline{e}_1^H B) \\ \cdot \\ \cdot \\ (e_n)^T \otimes (\underline{e}_n^H B) \end{bmatrix} \tag{15}$$

and the vector $\beta(r)$ is

$$\beta(r) = \begin{bmatrix} -\underline{e}_n{}^H\ BL(I-rDL)^{-2}\ Ce_1 \\ \cdot \\ \cdot \\ -\underline{e}_n{}^H\ BL(I-rDL)^{-2}\ Ce_n \end{bmatrix} \tag{16}$$

Note that the block-diagonal structure of $K_d(r)$ fixes many of the entries of $\text{vec}(\frac{dK_d(r)}{dr})$ at zero. After removing such entries and the corresponding columns of $\hat{P}(r)$, it follows that

$$P(r) \begin{bmatrix} \text{vec } (\frac{d}{dr} K_{d1}) \\ \cdot \\ \cdot \\ \text{vec } (\frac{d}{dr} K_{dN}) \end{bmatrix} = \beta(r) \tag{17}$$

A flow chart of the computational algorithm is given in Figure 4.

CONDITIONS FOR EXISTENCE

Equations (12, 14, 17) form the basis of the computational algorithm. The generic existence of a real solution to equation (17) is guaranteed under the following conditions:

i) Each component/subsystem is controllable and observable
ii) At most one component has an augmented state space which is of odd order (component and compensator taken as a whole)
iii) Complex eigenvalues must be distinct and in conjugate pairs.

In the CCM setting condition (i) is necessary for global as well as decentralized eigenvalue assignment. Condition (ii) and (iii) guarantee that the feedback gains are real. Note that (ii) implies that "dynamic compensation" is in general necessary to assign an arbitrary spectrum by decentralized control even though all the local states are available in each component. This is a general result unrelated to the computational algorithm.

EXAMPLE

Let the CCM of a 3-component system be given as follows

Component 1:

$$\begin{aligned} [\dot{x}_1^1] &= [1][x_1^1] + [1][a_1^1] \\ [b_1^1] &= [1][x_1^1] + [0][a_1^1] \end{aligned} \tag{18}$$

Component 2:

$$\begin{bmatrix} \dot{x}_1^2 \\ \dot{x}_2^2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} [a_1^2]$$

$$\begin{bmatrix} b_1^2 \\ b_2^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1^2 \\ x_2^w \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} [a_1^2] \tag{19}$$

Component 3:

$$\begin{bmatrix}\dot{x}_1^3\\ \dot{x}_2^3\\ \dot{x}_3^3\end{bmatrix} = \begin{bmatrix}0&0&1\\1&0&0\\0&1&0\end{bmatrix}\begin{bmatrix}x_1^3\\ x_2^3\\ x_3^3\end{bmatrix} + \begin{bmatrix}1&0\\0&1\\0&0\end{bmatrix}\begin{bmatrix}a_1^3\\ a_2^3\end{bmatrix}$$

$$\begin{bmatrix}b_1^3\\ b_2^3\\ b_3^3\end{bmatrix} = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}x_1^3\\ x_2^3\\ x_3^3\end{bmatrix} + \begin{bmatrix}0&0\\0&0\\0&0\end{bmatrix}\begin{bmatrix}a_1^3\\ a_2^3\end{bmatrix} \tag{20}$$

Connection matrix:

$$L = \begin{bmatrix}0&0&0&0&0&1\\1&0&0&0&0&0\\0&1&-1&0&0&0\\0&0&0&0&0&0\end{bmatrix} \tag{21}$$

Since there are 2 odd-order components, component 1 or 3 must be extended by first order dynamic compensation. Suppose a compensator with eigenvalue at -2 is incorporated into component 1, then this new component is characterized by

$$\begin{bmatrix}\dot{x}_1^1\\ \dot{x}_2^1\end{bmatrix} = \begin{bmatrix}0&1\\0&-2\end{bmatrix}\begin{bmatrix}x_1^1\\ x_2^1\end{bmatrix} + \begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}a_1^1\\ a_2^1\end{bmatrix}$$

$$\begin{bmatrix}b_1^1\\ b_2^1\end{bmatrix} = \begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}x_1^1\\ x_2^1\end{bmatrix} + \begin{bmatrix}0&0\\0&0\end{bmatrix}\begin{bmatrix}a_1^1\\ a_2^1\end{bmatrix} \tag{22}$$

In which case the connection matrix becomes

$$L = \begin{bmatrix}0&0&0&0&0&0&1\\0&0&0&0&0&0&0\\1&0&0&0&0&0&0\\0&0&1&-1&0&0&0\\0&0&0&0&0&0&0\end{bmatrix} \tag{23}$$

The components have respectively eigenvalues $\sigma_1 = \{1, -2\}$,

$\sigma_2 = \{-.5 \pm j\ .866\}$ and $\sigma_3 = \{-.5 \pm j\ .866,\ 1\}$. Suppose the objective is to assign the composite spectrum $\Lambda = \{-2 \pm j,\ -1 \pm j,\ -.5 \pm .5j,\ -3\}$. in the first step Λ is partitioned as $\Lambda = \Lambda_1 \uplus \Lambda_2 \uplus \Lambda_3$ where $\Lambda_1 = \{-2+-j\}$, $\Lambda_2 = \{-1+-j\}$, $\Lambda_3 = \{-.5 \pm .5j,\ -3\}$. The spectra Λ_i are assigned to the components by the respective initial feedback matrices:

$$K_{d1}(0) = \frac{1}{3}\begin{bmatrix} -10 & 1 \\ -10 & 1 \end{bmatrix}$$

$$K_{d2}(0) = \begin{bmatrix} 0 & -1 \end{bmatrix} \tag{24}$$

$$K_{d3}(0) = \begin{bmatrix} -1.5 & -2.5 & -1 \\ -1.5 & -2.5 & -1 \end{bmatrix}$$

Implementation of the above described algorithm resulted in the decentralized gains

$$K_{d1}(1) = \begin{bmatrix} -3.86 & .420 \\ -4.19 & .860 \end{bmatrix}$$

$$K_{d2}(1) = \begin{bmatrix} .00831 & -.982 \end{bmatrix}$$

$$K_{d3}(1) = \begin{bmatrix} -.889 & -12.07 & -.746 \\ -1.08 & -3.13 & -1.68 \end{bmatrix}$$

On CDC 6500, the computation time required to obtain $K_d(1)$ from $K_d(0)$ is about 15 seconds.

CONCLUSIONS

A continuations approach has been used to design local controllers for assigning the eigenvalues of a composite system modeled by the Component Connection Method. It is not necessary to transform the state variables of the system which facilitates the utilization of the algorithm. It is possible to apply the same idea to decentralized output feedback control [5]. To the authors' knowledge such algorithms were not available before.

ACKNOWLEDGMENT

This work was supported by DOE Emergency State Contract No. DE-AC01-79ET29365.

REFERENCES

[1] Wang, S. H. and Davison, E. J., "On the stabilization of decentralized control systems," IEEE Trans. Auto. Control, Vol. AC-18, October 1973, pp. 473-478.

[2] Corfmat, J. P. and Morse, A. S., "Decentralized control of linear multivariable systems," Automatica, Vol. 12, September 1976, pp. 479-495.

[3] Saeks, R., "On the decentralized control of interconnected dynamical systems," IEEE Trans. Auto. Control, Vol. AC-24, April 1979, pp. 269-271.

[4] Saeks, R. and DeCarlo, R., Interconnected dynamical systems, Marcel Dekker, to appear

[5] Lefebvre S., Richter S, and DeCarlo R., "Decentralized control of linear interconnected multivariable systems," 1980 Conference on Decision and Control, December 1980.

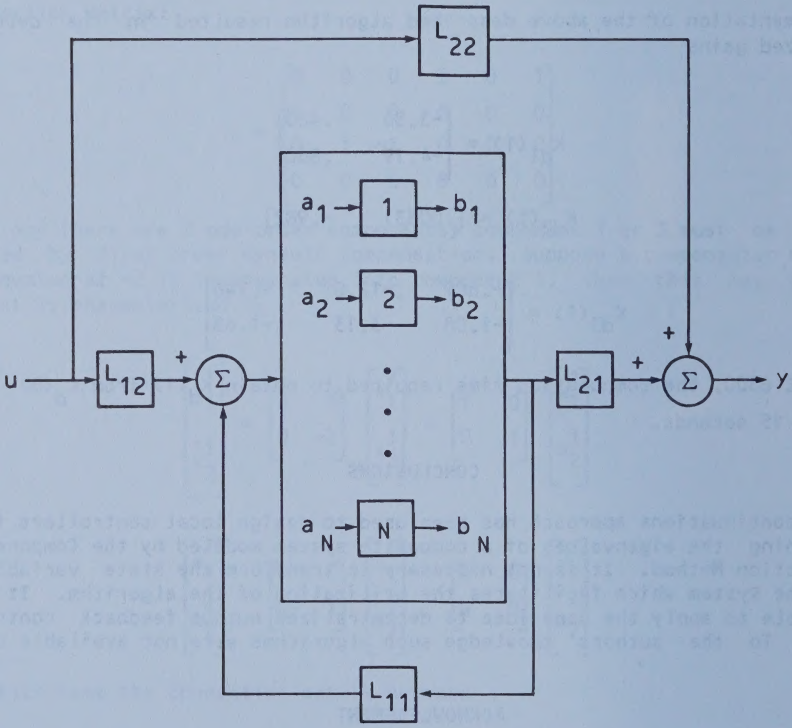


Figure 1. Detailed block diagram of CCM.

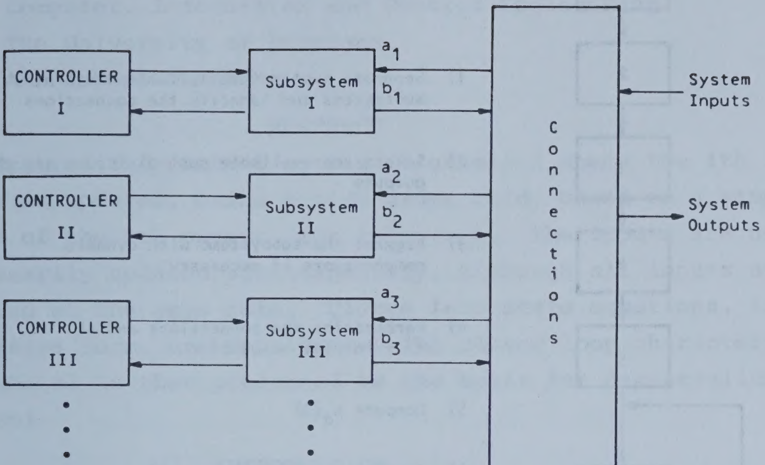

CONTROLLER I
CONTROLLER II
CONTROLLER III
Subsystem I
Subsystem II
Subsystem III
a_1
b_1
a_2
b_2
a_3
b_3
Connections
System Inputs
System Outputs

Figure 2. Block diagram of decentralized control strategy in CCM context.

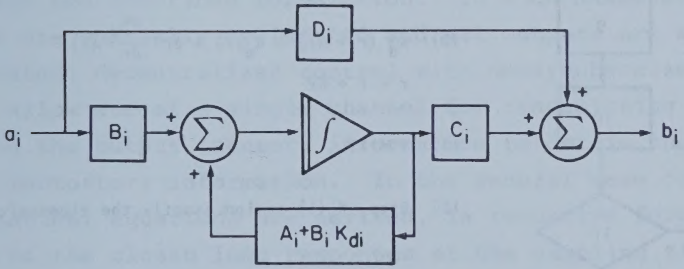
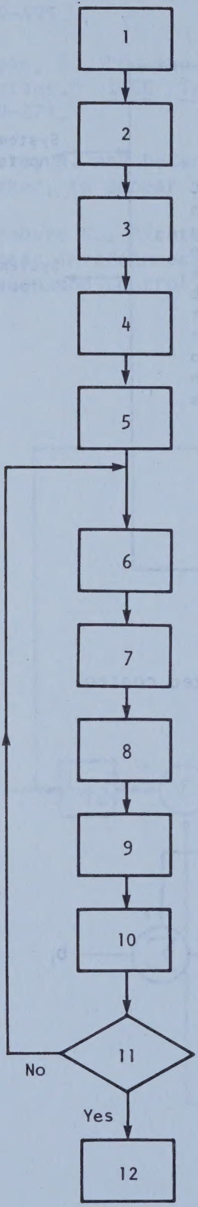

D_i
a_i
B_i
+
+
$\sum$
$\int$
C_i
+
+
$\sum$
b_i
$A_i+B_i K_{di}$

Figure 3. Block diagram of compensated component.



1) Separate system into suitable disjoint subsystems and identify the connections

2) Select the available control inputs and the outputs

3) Augment the subsystems with dynamic compensators if necessary

4) Parametrize the connections and rL
$r \leftarrow$ initial value of r .

5) Compute $K_d(0)$

6) Compute the eigenvectors of

$$A + BrL\,(I - rDL)^{-1}C + BK_d(r)C$$

7) Compute $\beta(r)$ and $P(r)$

8) Compute vec $(\frac{d}{dr}\,K_d) = P^{-R}(r)\,\beta(r)$ where

$-R$ stands for right inverse

9) Select an appropriate step size Δr

10) $K_d\,(r + \Delta r) \leftarrow K_d(r) + \Delta r\,\dfrac{dK_d}{dr}\,(r)$

$r \leftarrow r + \Delta r$

11) $r = 1$?

12) Stop, $K_d(1)$ assigns exactly the eigenvalues

Figure 4. Flow chart outlining decentralized control algorithm.

ON SAMPLED DATA DECENTRALIZED CONTROL

N. HARRIS MCCLAMROCH
Computer, Information and Control Engineering
The University of Michigan
Ann Arbor, Michigan

ABSTRACT

A decentralized control system is examined where the ith input value is updated, using a zero order hold, based on a sampled value of the ith output, for i=1,...,p. The inputs are not necessarily updated simultaneously, although all inputs are updated at the same rate. Closed loop state equations, in recursive form, are developed. The closed loop characteristic polynomial is then presented as the basis for decentralized control.

INTRODUCTION

A decentralized control problem is examined where the ith input value is updated, using a zero order hold, based on a sampled value of the ith output value, for i=1,...,p. The output sample times (and the input update times) are not synchronized throughout the cycle time; but all input/output pairs are sampled (and updated) at the same rate. A class of decentralized control systems is briefly presented as motivation for the described formulation. In such examples, all inputs are spatially collocated and all outputs are spatially collocated; decentralized control with nonsynchronized sampling might allow use of a single channel for transmitting and processing the output (sensor) information to obtain the required input (actuator) information. In the general case considered, mathematical equations are derived, in recursive form, which describe the closed loop responses at the sampling times. The closed loop characteristic polynomial is easily derived. The special case where all output samplings (and control updates) occur simultaneously is considered and contrasted with the more general situation.

THE SAMPLED DATA MODEL

A standard linear decentralized control system, as described in [4,6], is given in the form

$$\frac{dx}{dt} = Ax + \sum_{i=1}^{p} B_i u_i \ ,$$

$$y_i = C_i x \quad , \quad i=1,\ldots,p. \tag{1}$$

There are p input variables $u_1,\ldots,u_p$ and p output variables $y_1,\ldots,y_p$, and x is the state vector. This general model includes, as special cases, the linear models considered in [5]. It is assumed that the ith input value u_i is updated, using a zero order hold, based on an output sampled value y_i, for each i=1,...,p. The p output samplings (and the p control updates) are not assumed to occur simultaneously, although they are assumed to occur at the same rate. If this sampling rate (and control update rate) is $2\pi/\tau$ for $\tau>0$ then the inputs satisfy

$$u_i(t) = u_i(k\tau+\tau_i) \text{ for } k\tau+\tau_i \leqslant t < k\tau+\tau+\tau_i, \quad i=1,\ldots,p;\ k=0,1,\ldots \tag{2}$$

and the outputs satisfy

$$y_i(k\tau+\tau_i) = C_i x(k\tau+\tau_i), \quad i=1,\ldots,p;\ k=0,1,\ldots \tag{3}$$

Without loss of generality, it is convenient to assume that the input/output pairs are ordered so that

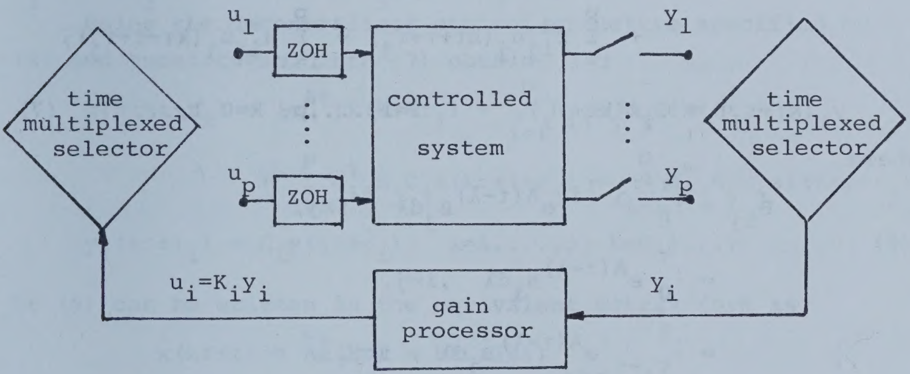
$$0<\tau_1<\tau_2 \ \ldots \ <\tau_p<\tau \ . \tag{4}$$

The decentralized control problem, considered in this paper, consists of choosing feedback gain matrices $K_1,\ldots,K_p$, for the decentralized controller

$$u_i(k\tau+\tau_i) = K_i y_i(k\tau+\tau_i) \ , \quad i=1,\ldots,p;\ k=0,1,\ldots \tag{5}$$

so that the resulting closed loop system, as defined by equations (1), (2), (3), (5), is asymptotically stable. A similar control problem, with a sampling constraint like (4), but without the requirement of decentralized control, has been studied previously by the author in [3].

Before proceeding with a formal mathematical development, it may be useful to mention a class of control systems which has served as motivation for the above formulation. Suppose that the system to be controlled is such that the p input variables (implemented by p physical actuators) are spatially collocated and that the p output variables (implemented by p physical sensors) are spatially collocated. Use of decentralized control may be feasible if the ith output y_i depends only "weakly" on the jth input u_j, for $j \neq i$. By using decentralized control as in (5), with nonsynchronized sampling times as in (4), only one channel is required for transmitting input/output information and for processing the output sampled values to obtain the input update values. Thus such a closed loop control system is efficient, from the viewpoint of transmission and processing of input/output information. A schematic of such a sampled data feedback control system is indicated below.



Although the above illustration has served as motivation for this work, the general model described by (1), (2), (3), (5) is applicable to any sampled data decentralized control system where the p output samplings (or p control updates) are not synchronized.

RECURSIVE STATE MODELS

As is usual in describing sampled data feedback systems, it is desirable to develop a recursive model which relates the values of the state at sampling times throughout one cycle to values of the state at sampling times during the previous cycle and previous input values at the sampling times. Such procedures are well known and are described in [2]. In this particular case, since the sampling times are not synchronized according to (4), a recursive equation is developed in terms of the state values $x(k\tau+\tau_1),\ldots,x(k\tau+\tau_p)$, $k=0,1,\ldots$

From equation (1) it follows that

$$x(k\tau+\tau+\tau_i) = [e^{A\tau}]x(k\tau+\tau_i) + \int_0^{\tau} e^{A(\tau-\lambda)} \sum_{i=1}^{p} B_i u_i(k\tau+\tau_i+\lambda)\,d\lambda, \quad i=1,\ldots,p;\ k=0,1\ldots \qquad (6)$$

Using equation (2) obtain

$$\begin{aligned} x(k\tau+\tau+\tau_i) &= [e^{A\tau}]x(k\tau+\tau_i) + \sum_{i=1}^{p} B_{ij}u_j(k\tau+\tau_j) \\ &\quad + \sum_{i=1}^{p} B^{\ell}_{ij}u_j(k\tau+\tau+\tau_j) + \sum_{i=1}^{p} B^{u}_{ij}u_j(k\tau-\tau+\tau_j), \end{aligned}$$

$$y_i(k\tau+\tau_i) = C_i x(k\tau+\tau_i), \qquad i=1,\ldots,p;\ k=0,1,\ldots \qquad (7)$$

where

$$\begin{aligned} B_{ij} &= \int_0^{\tau+\tau_j-\tau_i} e^{A(\tau-\lambda)}B_j\,d\lambda\ , \ i>j, \\ &= \int_0^{\tau} e^{A(\tau-\lambda)}B_i\,d\lambda\ , i=j, \\ &= \int_{\tau_j-\tau_i}^{\tau} e^{A(\tau-\lambda)}B_j\,d\lambda\ , \ i<j, \\ B^{\ell}_{ij} &= \int_{\tau+\tau_j-\tau_i}^{\tau} e^{A(\tau-\lambda)}B_j\,d\lambda\ , \ i>j, \\ &= 0, \quad i\leq j, \\ B^{u}_{ij} &= \int_0^{\tau_j-\tau_i} e^{A(\tau-\lambda)}B_j\,d\lambda\ , \ i<j, \\ &= 0, \quad i\geq j. \end{aligned}$$

The above equations (7) can be written more compactly in a matrix form by introducing the partitioned vectors

$$\hat{x}(k\tau) = \begin{bmatrix} x(k\tau+\tau_1) \\ \vdots \\ x(k\tau+\tau_p) \end{bmatrix}, \ \hat{u}(k\tau) = \begin{bmatrix} u_1(k\tau+\tau_1) \\ \vdots \\ u_p(k\tau+\tau_p) \end{bmatrix}, \ \hat{y}(k\tau) = \begin{bmatrix} y_1(k\tau+\tau_1) \\ \vdots \\ y_p(k\tau+\tau_p) \end{bmatrix}$$

so that (7) can be written in the matrix form

$$\begin{aligned} \hat{x}(k\tau+\tau) &= \hat{A}\hat{x}(k\tau) + B^u\hat{u}(k\tau-\tau) + \hat{B}\hat{u}(k\tau) + B^\ell\hat{u}(k\tau+\tau), \\ \hat{y}(k\tau) &= \hat{C}\hat{x}(k\tau), \quad k=0,1,\ldots \end{aligned} \tag{8}$$

The following partitioned matrices have been introduced:

$\hat{A} = \text{diag}[e^{A\tau},\ldots,e^{A\tau}]$ is a block diagonal matrix;

$B^u = [B^u_{ij}]$ is an upper triangular matrix;

$B^\ell = [B^\ell_{ij}]$ is a lower triangular matrix;

$\hat{B} = [B_{ij}]$ is a matrix with no special structure;

$\hat{C} = \text{diag}[C_1,\ldots,C_p]$ is a block diagonal matrix.

The triangular structure of the matrices B^u and B^ℓ is significant, since it follows that $x(k\tau+\tau+\tau_i)$ <u>does</u> <u>not</u> <u>depend</u> on $u_j(k\tau-\tau+\tau_j)$, $j<i$, or on $u_j(k\tau+\tau+\tau_j)$, $j>i$.

Using the decentralized control structure specified by (5) and substituting into (7) obtain

$$\begin{aligned} x(k\tau+\tau+\tau_i) &= [e^{A\tau}]x(k\tau+\tau_i) + \sum_{i=1}^{p} B_{ij}K_jC_jx(k\tau+\tau_j) \\ &\quad + \sum_{i=1}^{p} B^\ell_{ij}K_jC_jx(k\tau+\tau+\tau_j) + \sum_{i=1}^{p} B^u_{ij}K_jC_jx(k\tau-\tau+\tau_j), \\ y_i(k\tau+\tau_i) &= C_ix(j\tau+\tau_i), \quad i=1,\ldots,p; \ k=0,1,\ldots \end{aligned} \tag{9}$$

Or (9) can be written in the equivalent matrix form as

$$\begin{aligned} \hat{x}(k\tau+\tau) &= \hat{A}\hat{x}(k\tau) + \hat{B}\hat{K}\hat{C}\hat{x}(k\tau) \\ &\quad + B^u\hat{K}\hat{C}\hat{x}(k\tau-\tau) + B^\ell\hat{K}\hat{C}\hat{x}(k\tau+\tau), \\ \hat{y}(k\tau) &= \hat{C}\hat{x}(k\tau), \qquad k = 0,1,\ldots \end{aligned} \tag{10}$$

where $\hat{K} = \text{diag}[K_1,\ldots,K_p]$ is a block diagonal matrix. Note that $B^u\hat{K}\hat{C}$ is upper triangular and $B^\ell\hat{K}\hat{C}$ is lower triangular. Thus, the state equations (9), or equivalently (10), are well posed in the following sense: given $x(-\tau+\tau_1),\ldots,x(-\tau+\tau_p),x(\tau_1),\ldots,x(\tau_p)$, there is a unique solution $x(k\tau+\tau_i)$ of (9), or (10), for each $i=1,\ldots,p$, for $k=0,1,\ldots$

DESIGN METHODOLOGY

The state equations (9), or (10), can naturally serve as a basis for choosing the decentralized gain matrices $K_1,\ldots,K_p$. In particular the closed loop characteristic polynomial is easily obtained as

$$\det[(I-B^{\ell}\hat{K}\hat{C})z^2 - (\hat{A}+\hat{B}\hat{K}\hat{C})z - B^{u}\hat{K}\hat{C}]. \tag{11}$$

Since $B^{\ell}\hat{K}\hat{C}$ is lower triangular, the matrix $I-B^{\ell}\hat{K}\hat{C}$ is necessarily invertible; thus the closed loop characteristic polynomial is of degree 2np. The factor of two in this degree is due to the presence of "delays" in (10); this occurs as a direct consequence of the constraints (4).

In order to guarantee that the closed loop system (10) is stable the decentralized control gain matrices $K_1,\ldots,K_p$ should be chosen so that the characteristic zeros of (11) all lie inside the unit circle in the complex plane. Since the characteristic zeros are extremely complicated functions of the gains it is not easy to develop explicit conditions for stabilization. In fact, this stabilization problem is significantly more difficult that the decentralized stabilization problem considered in [1,6].

It is interesting to make a contrast with the special case where the samplers (and control updates) are synchronized, i.e. $\tau_1=\tau_2 = \ldots =\tau_p=0$. In this case it follows that

$$B^{\ell} = B^{u} = 0$$

so that the closed loop characteristic polynomial is

$$\det[z^2 I - (\hat{A}+\hat{B}\hat{K}\hat{C})z]. \tag{12}$$

Thus there are np characteristic zeros which satisfy

$$\det[zI - (\hat{A}+\hat{B}\hat{K}\hat{C})] \tag{13}$$

and np characteristic zeros at zero. And stabilization of the closed loop sampled data system in this special case is thus exactly of the form considered in [1,6]. Moreover, it follows that if the np characteristic zeros of (13) lie inside the unit circle, then, in the general case described by (10), if τ_p is sufficiently small the 2np characteristic zeros of (11) must also lie inside the unit circle. This observation naturally suggests a design methodology based on equation (13) if τ_p is small. Formal design procedures, otherwise, are not obvious.

CONCLUSIONS

A certain class of sampled data decentralized control systems has been described; the interesting feature is the constraint (4). Recursive closed loop state equations (10) have been developed; the closed loop characteristic polynomial (11) has been presented as a possible basis for choosing the decentralized feedback gain matrices.

REFERENCES

1. J.P. Corfmat and A.S. Morse, "Decentralized Control of Linear Multivariable Systems," Automatica, 12, 1976, 479-495.
2. G.F. Franklin and J.D. Powell, Digital Control of Dynamic Systems, Addison-Wesley, 1980.
3. N.H. McClamroch, "Control of a Linear Sampled-Data System Using Round-Robin Output Feedback," IEEE Transactions on Automatic Control, AC-25, 1980, 298-299.
4. N.R. Sandell, et.al., "Survey of Decentralized Control Methods for Large Scale Systems," IEEE Transactions on Automatic Control, AC-23, 1978, 108-128.
5. D.D. Siljak, Large-Scale Dynamic Systems, North-Holland, 1978.
6. S. Wang and E.J. Davison, "On the Stabilization of Decentralized Control Systems," IEEE Transactions on Automatic Control, AC-18, 1973, 473-478.

SOME MODIFICATIONS TO THE MEDIAN FILTER PROCESS AND THEIR PROPERTIES[†]

T. A. NODES AND N. C. GALLAGHER, JR.
School of Electrical Engineering
Purdue University
W. Lafayette, IN 47907

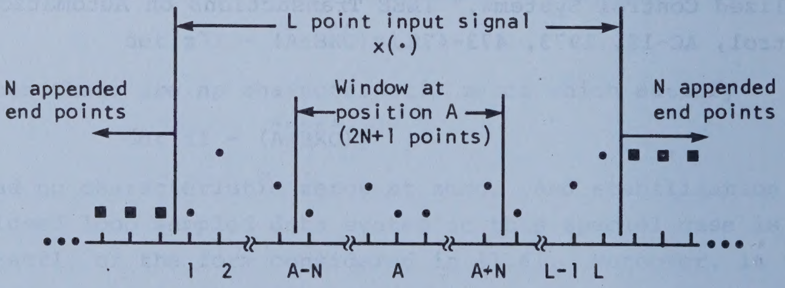
ABSTRACT

Some modifications of the median filter are given and their properties are derived. In addition, some results for standard median filters are given. It is shown that for non median nth ranked-order operations, repeated application of the operation will reduce any signal to a constant. Also, it is proved that the output of a recursive median filter is invarient to subsequence passes by the same filter.

I. INTRODUCTION

Median filtering, a method of signal processing which is easily implemented on a digital computer, has been used with success in many applications. These applications include picture processing and speech processing[1,2,3,4] where it is employed to smooth the signal. Further potentially useful properties can be obtained from slight modifications of the median process. We have investigated several such modifications and present the properties of two of them. In section II, we look at the nth ranked-order operation, which is a generalization of the median process. In section III, we study the recursive median operation, which incorporates previous output values into the median decision process. Finally, in section IV we introduce some other possible modifications to median filters. First, however, a review of the standard median filter is in order.

Median filtering is a discrete time process in which a 2N+1 points wide window is stepped across an input signal (see Fig. 1). At each step, the points inside the window are ranked according to their values, and the median value (mid-point) of the ranked set is taken as the output value of the filter for each window position. At both ends of the signal, N end points are appended to allow the filter to reach the edges of the signal.



The output of the median filter, $Y(A)$ is given by

$$Y(A) = \text{the median value of } \{x(A-N),\ldots,x(A-1),x(A),x(A+1),\ldots,x(A+N)\}$$

Figure 1: The Median Filter

[†] The authors gratefully acknowledge the support of the Air Force Office of Scientific Research under Grant AFOSR 78-3605.

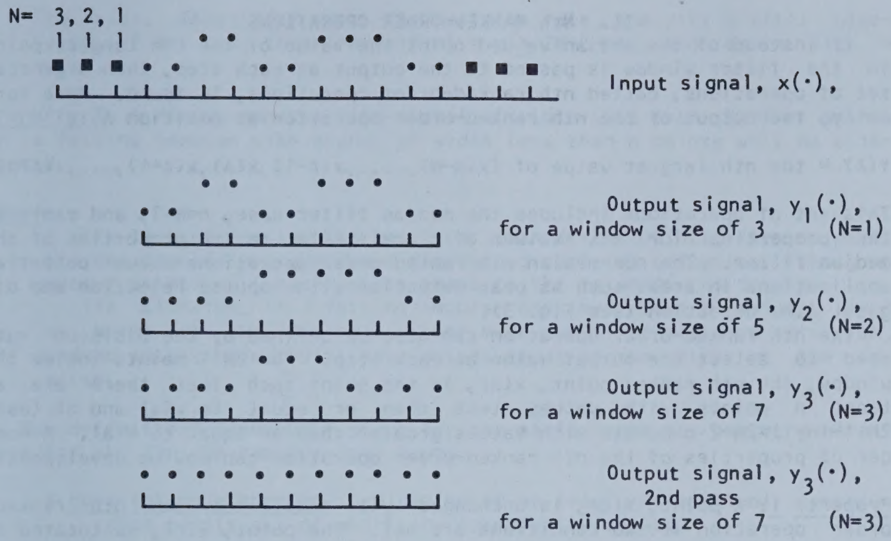

N= 3, 2, 1
Input signal, $x(\cdot)$,
Output signal, $y_1(\cdot)$, for a window size of 3 (N=1)
Output signal, $y_2(\cdot)$, for a window size of 5 (N=2)
Output signal, $y_3(\cdot)$, for a window size of 7 (N=3)
Output signal, $y_3(\cdot)$, 2nd pass for a window size of 7 (N=3)

Figure 2: Effects of window size on a median filtered signal

The value of the front endpoints is equal to the value of the first point of the signal, and the value of the rear endpoints is equal to last point of the signal. As an example of this process, consider Fig. 2. Here, a binary signal of length eleven (the ■'s represent the appended endpoints) is median filtered by three different window widths N = 1 (2N+1=3), N = 2 (2N+1=5), and N = 3 (2N+1=7). Notice, for the N=1 case, the signal is unperturbed, while for the N=2 and N=3 cases, the amount of structure in the signal is reduced. A number of signal structures which can be used to define the properties of median filters, can now be defined.

A constant neighborhood is a region of at least N+1 consecutive points all of which are identically valued.

An edge is a monotonically rising or falling set of points surrounded on both sides by constant neighborhoods.

An impulse is a set of N or less points whose values are different from the surrounding regions and whose surrounding regions are identically valued constant neighborhoods.

A root is a signal which is not modified by filtering.

Gallagher and Wise[5] have shown that, while impulses are eliminated by median filtering, constant neighborhoods and edges are unperturbed, and in fact, only signals composed solely of constant neighborhoods and edges are roots to the median filter. Again referring to Fig. 2, note that the signal is a root of the N=1 median filter but not for filters with N greater than one. However, after one pass of the N=2 filter or two passes of the N=3 filter the resulting outputs are roots of their respective filters. In fact, Gallagher and Wise have also proven that any signal of length L is reduced to its root after at most $\frac{1}{2}\cdot(L-2)$ successive passes by any median filter. Furthermore, any root of a median filter with a particular window size is also a root of any median filter with a smaller window size.

II. Nth RANKED-ORDER OPERATIONS

If instead of the median valued point the value of the nth largest point in the filter window is passed to the output at each step, then a general set of operations, called nth ranked-order operations, is found. More formally, the output of the nth ranked-order operation at position A is

$$Y(A) = \text{the nth largest value of } \{x(A-N),\ldots,x(A-1),x(A),x(A+1),\ldots,x(A+N)\}$$

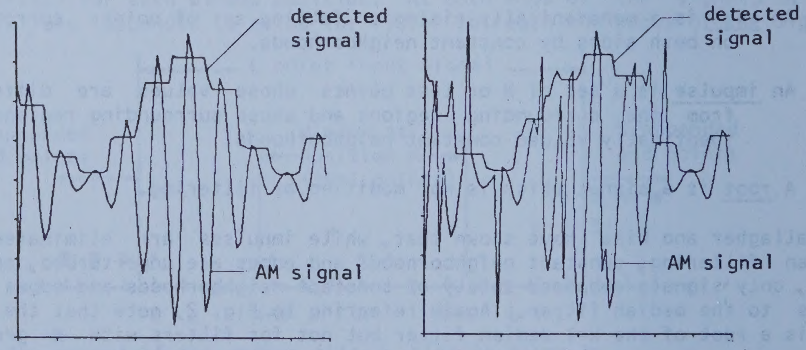
This set of operations includes the median filter case, $n=N+1$, and many of the properties for all values of n are similar to the properties of the median filter. The non-median nth ranked-order operations have potential applications in areas such as peak detection with impulse rejection and digital A.M. detection (see Fig. 3).

The nth ranked-order operation can also be defined by the decision rule used to select the output value at each step. For 2N+1 points inside the window, the nth ranked point, $x(a)$, is the point such that there are at least n points with values less than or equal to $x(a)$ and at least $2N+1-(n-1)=2N+2-n$ points with values greater than or equal to $x(a)$. A number of properties of the nth ranked-order operation can now be developed.

Property 1: A point, $X(t)$, is unchanged ($y(t) = x(t)$) by an nth ranked-order operation if two conditions are met. The point, $x(t)$, is located in a constant region, and $x(t)$'s position is restricted to $b+N-a \leq t \leq c-[|N+1-n|+a]$ where a is any nonnegative integer of value less than $N+1-|N+1-n|$ and b and c are the positions of the two endpoints of the constant region

Proof:

Assume that the two conditions given above are met. Now, let $a = 0$. The constant region must now extend to at least N points left (decreasing t) of $x(t)$ and $|N+1-n|$ points right of $x(t)$ for a total of at least $1+N+|N+1+n|$ points of value $x(t)$ inside the window. Furthermore, if $a \neq 0$, then the constant region will extend 'a' fewer points to the left of $x(t)$ but 'a' more points to the right, thus, maintaining a total of at least $N+1+|N+1+n|$ constant valued points inside the window. This means that if $N+1 \geq n$ then at least $1+N+|N+1-n| = 2N+2-n$ ($\geq n$) points inside the window have values equal



(a) (b)

Figure 3: A.M. Detection of a 5KHz tone on a 31KHz carrier and sampled at 250KHz using an 8th ranked-order operation with a window size of 9
(a) original signal (b) signal corrupted with impulse noise

to $x(t)$. Thus, $x(t)$ meets the decision rule, and $y(t) = x(t)$. Likewise, if $N+1 < n$, then $1+N+|N+1-n| = n(\geq 2N+2-n)$, and again $y(t) = x(t)$.

Property 2: A rising impulse like signal of width less than $2N+2-n$ points or a falling impulse like signal of width less than n points will be eliminated.

Proof:

i) If a rising impulse has fewer than $2N+2-n$ points, then no point of the impulse can ever meet the second decision criterion. Thus, no output points will have values equal to the value of the impulse.

ii) Likewise, if a falling impulse has fewer than n points, then no point of the impulse can ever meet the first decision criterion, and no output points will have values equal to the value of the impulse.

The definitions previously given for the median case may now be generalized for all the nth ranked-order cases.

A constant neighborhood is a region of at least $N+1+|N+1-n|$ consecutive points all of which are identically valued.

An impulse is a set of points whose values are different from the surrounding regions and whose surrounding regions are identically valued constant neighborhoods. If the values of this set of points are greater than the surrounding neighborhoods, then the impulse contains less than $2N+2-n$ points, and if the values of the impulse are less than the surrounding regions, then the impulse contains less than n points.

The definitions for the edge and the root are unchanged. Note that, property 2 can be restated as "impulses are eliminated by nth order operations". Using these definitions, further properties can be developed. Due to lack of space, however, many of these properties are presented without proof.

Property 3: Upon each pass of an nth ranked-order operation, every edge of a signal will be moved to the left (advanced) by

$$\mathrm{sgn}[\mathrm{edge}]\cdot(n-N-1) \text{ points}$$

$$\text{where} \quad \mathrm{sgn}[\mathrm{edge}] = \begin{cases} +1 & \text{if } x(t) \leq x(t+1) \quad \text{For } t \text{ ranging over all} \\ -1 & \text{if } x(t) \geq x(t+1) \quad \text{positions in the edge} \end{cases}$$

Property 4:

Any constant region of $2N+2-n$ or more points surrounded by constant neighborhoods of lesser values will be changed in width by $2\cdot(n-N-1)$ points after being passed through an nth ranked-order operator.

Any constant region of n or more points surrounded by constant neighborhoods of greater values will after being operated on be changed in width by $2\cdot(N+1-n)$ points.

As can be seen from the above properties, for n greater than $N+1$ the maximum valued signal segment (or the minimum if n is less than $N+1$) which is not an impulse tends to expand its coverage with each pass of a non-median operator. Thus, under repeated operations, a signal tends to be reduced to a constant. That this is true for any signal is shown in the following properties.

Property 5: Only constant signals are invariant to nth ranked-order operations if n is not equal to N+1.

Property 6: If n is not equal to N+1, then repeated passes of an nth ranked-order process will reduce any finite length signal to a constant.

The output of an nth ranked-order operation at position Z is not influenced by input points more than N points ahead (>Z+N) or N points behind (<Z-N) Z. This suggests a method by which long signals could be segmented and the ranked-order operations on each segment carried out in parallel.

- i) Append the start and stop points as usual
- ii) Divide the signal into overlapping segments. Each overlap is 2N+1 elements wide.
- iii) Perform the normal nth ranked-order operation independently on each segement.
- iv) After each operation replace the last N points of each segment (except the last segment) with the N+2 through the 2N+1 points of the following segment. Also, replace the first N element of each signal segment (except the first segment) with the elements from the 2N+1 through N+2 positions preceding the end of the prior signal segment.

Now, the signal is the same as it would be had the processing been done before the segmentation. Thus, further processing can now be done, or the segments can be recombined to form the final output signal.

A signal may be formed from independent identically distributed, iid, sample points of a random process. Such a signal would be formed if white noise were sampled to form the input signal. For this type of signal, results from order statistics[6] may be used to obtain the first order distribution, $F_y(\cdot)$, and the density, $f_y(\cdot)$, of the output of an nth ranked-order operation. If the distribution, $F_x(\cdot)$, and the density, $f_x(\cdot)$, of the input are known, then $f_y(\cdot)$ and $F_y(\cdot)$ are given by

$$f_y(x) = \frac{(2N+1)!}{(n-1)!1!(2N+1-n)!}\left[F_x^{n-1}(x)(1-F_x(x))^{(2N+1-n)}\ f_x(x)\right] \quad \text{property \#7}$$

$$F_y(x) = \sum_{K=n}^{2N+1} \frac{(2N+1)!}{K!(2N+1-K)!}\,F_x^K(x)\ (1-F_x(x))^{2N+1-K} \quad \text{property \#8}$$

where 2N+1 is the window size.

Kuhlmann and Wise[7] will present further statistical analysis of the median filtering of independent identically distributed random processes in the next paper. However, the above formulas can immediately be used to prove that the statistical median of an iid process is preserved under standard median filtering.

Property 9: A median filter, $x(\cdot) \rightarrow y(\cdot)$; with an input of iid sample points will transform the distribution of the input, $F_x(\cdot) \rightarrow F_y(\cdot)$, symmetrically about 0.5. That is, for any ℓ such that $F_x(\ell) \rightarrow F_y(\ell)$, then $(1-F_x(\ell)) \rightarrow (1-F_y(\ell))$.

Property 10: The statistical median of a signal of iid sample points is preserved upon median filtering, or given ℓ such that $F_x(\ell) = 0.5$, then $F_y(\ell) = 0.5$.

Also recall that if the density of the input, $f_x(\cdot)$, is symmetric, then the mean, $E_X\{\ \}$, and the median are equal. Therefore, by properties 9 and 10, the mean of an iid sample point signal whose density is symmetric is also preserved under median filtering. However, in general, the actual median point and the average of a particular signal will not be preserved.

Recursive Operations

Now consider replacing, at every step, the leftmost N points in the moving window with the previous N output points, and apply the same decision rule as was previously given for the nth ranked-order operation to obtain the next output value. This produces a recursive nth ranked-order operation which can be more formally stated as follows.

$Y(A)$ = the nth largest value of $\{Y(A-N),\ldots,Y(A-1),X(A),X(A+1),\ldots,X(A+N)\}$

Where $X(A)$ and $Y(A)$ are the values of the input and the output respectively at position A. The properties of these operations are similar to those of standard nth ranked-order operations. Most notably, they have the same set of roots.

Property 11: A signal is invariant to recursive filtering if and only if it is invariant to standard filtering.

Proof:

If a signal is invariant to an operation, $X(\cdot) \rightarrow Y(\cdot)$, then $X(k) = Y(k)$ for all k. Therefore, if a signal is invariant, then standard and recursive operations use the same points in the decision rule, and they must produce the same resulting signal.

However, the same signal will not in general reduce to the same root under recursive and standard operations. This is illustrated by an example for the median (n=N+1) filter case in Fig. 4. One may notice that under noisy conditions, the recursive filter tends to maintain a higher correlation between points in its output than does its non-recursive counterpart. This is further illustrated in figure 5 which compares the autocorrelation of the output for recursive and standard median filters with independant uniformly [0,1] distributed input points. These autocorrelation functions were obtained experimentally from a sequence of 2,200 random points. Thus, these filters may be useful in cases where more stringent filtering without a wider window is required.

One of the most interesting characteristics of the recursive operations

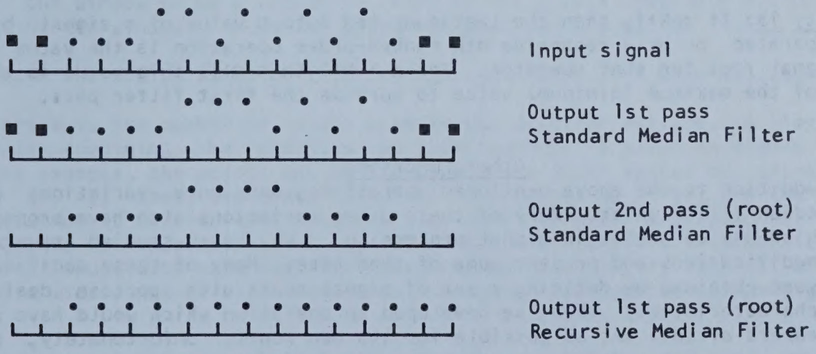


Figure 4: Recursive vs Standard Median filters with a window width of 5 (N=2)

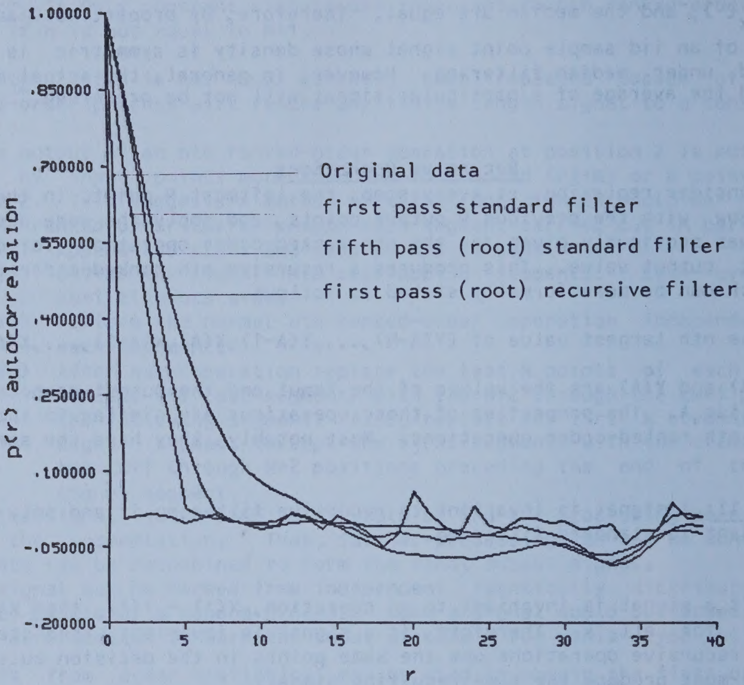


Figure 5: Autocorrelation function of standard and recursive median filters for a window width of five.

is that the root of a signal for a particular recursive process can always be found after the first pass of the operation. Recursive ranked-order operations are therefore potentially useful in areas, such as peak detection and coding operations, which require finding the root of a signal quickly. The following two properties prove this characteristic.

Property 12: Any signal will be reduced to a root after one pass of a recursive median filter (n=N+1).

Property 13: If n≠N+1, then the last computed output value of a signal being operated on by a recursive nth ranked-order operation is the value of the signal root for that operator. For n > N+1 (n < N+1) this value is the value of the maximum (minimum) value to survive the first filter pass.

Other Functions

In addition to the above mentioned operations, many more variations of the median filter exist. Many of these other variations also have properties which may be useful in signal processing. We have studied several such modifications and present some of them here. Many of these modifications were obtained by defining a set of signal roots with certain desirable characteristics; then, we developed an operation which would have as many members of this set as possible for its own roots. Unfortunately, we have not, as yet, found a systematic method of determining an operation which will have any particular set of roots. Nevertheless, this approach does appear to hold promise.

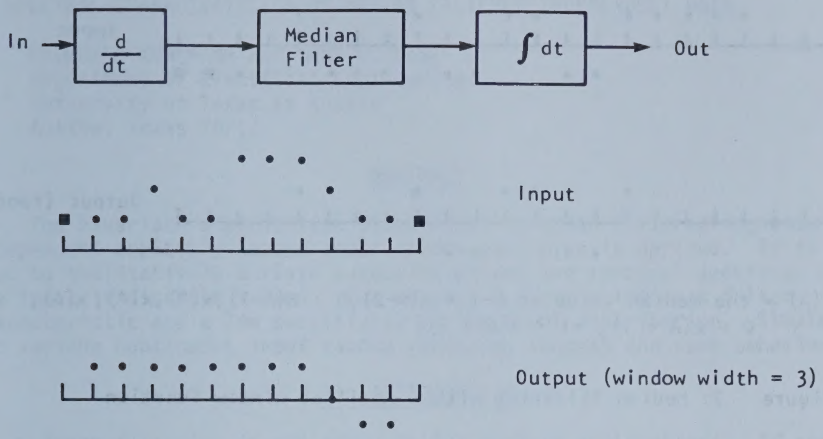


Figure 6: Example of a linear-median filter using a differentiator, integrator pair and a median filter with a window width of 3 (N=1)

One modification to the median filter, which Tukey [1] and Rabiner [2] have already utilized with promising results, is that of combining linear and median operations together. This allows one to greatly extend the number of available effects by utilizing some of the many linear operators whose properties are already well known. As an example of such an operation, consider figure 6. Here, a signal is differentiated, median filtered, and finally integrated. This operation has many of the same properties as a median filter alone. However, due to the differentiation, any slope of extent less than N+1 points will be seen by the median filter as an impulse and, thus, eliminated. Therefore, roots of this operation cannot contain sharp edges.

Another method of varying the median filter is to weight some positions of the window more heavily than others. This could be done by duplicating certain positions of the window. If the center position, for example, were to be weighted by three, then the output at position A would be given by

$$Y(A) = \text{the median value of } \{X(A-N), \ldots, X(A), X(A), X(A), \ldots, X(A+N)\}$$

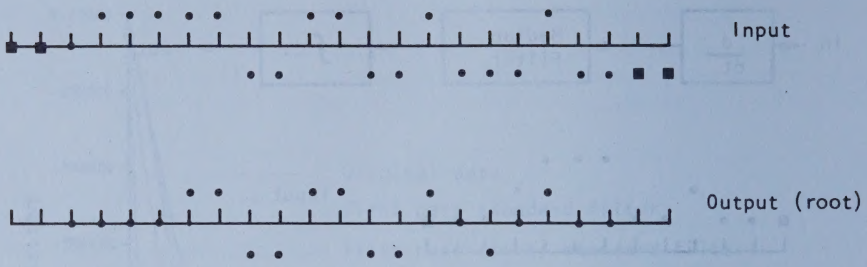
Yet, another modification would be to allow the value of a given position of the window to be a linear function of the points (possibly all of them) inside the window. Thus, the output at position A would be

$$Y(A) = \text{the median value of } \{f_1\{X(A-N), \ldots, X(A+N)\}, \ldots, f_m\{X(A-N), \ldots, X(A+N)\}\}$$

where m is the number of values used in the decision process. A simple example combining the previous two modifications is given in figure 7. In this example, the points inside the window are first scaled by either -1, 0, or +1; then, the center position is weighted by three, and the median operation is carried out. The roots of this operation are zero or those segments of periodicity 4 ($X(i) = X(i \pm 4)$) which are symmetric about zero. Thus, with some modifications, median type filters can be designed for a wide range of different roots, including some periodic type signals.

Conclusion

In this paper, we have examined several variants of the median filter. We have found that the set of nth ranked-order operations is a generaliza-



$Y(A)$ = the median value of $\{-1 \cdot x(A-2), 0 \cdot x(A-1), x(A), x(A), x(A),$
$0 \cdot x(A+1), -1 \cdot x(A+2)\}$

Figure 7: Median filtering with a modified window function

tion of the median filter, and that they all have many similar characteristics. However, the non-median operators will, after repeated passes, reduce any signal to a constant. In contrast, the recursive median process retains the same set of roots as a standard median filter, though the same signal may not reduce to the same root under both operations. However, the recursive median filter reduces any signal to a root in just one pass, and thus, may be useful where high speed root determination is required. We have also reviewed some examples of other types of modified median operations, including combined linear, median functions and filters with modified windows.

REFERENCES

1. J. W. Tukey, "Nonlinear (Nonsuperposable) Methods for Smoothing Data," in Cong. Rec., 1974 EASCON, p. 673.

2. L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a Nonlinear Smoothing Algorithm to Speech Processing," IEEE Trans. Acust., Speech, and Signal Processing, vol. ASSP-23, pp. 552-557, Dec. 1975.

3. N. S. Jayant, "Average and Median Based Smoothing Techniques for Improving Digital Speech Quality in the Presence of Transmission Errors," IEEE Trans. on Commun., vol. COM-24, pp. 1043-1045, Sept. 1976.

4. T. S. Huang, G. T. Yang, and G. Y. Tange, "A Fast Two-Dimensional Median Filtering Algorithm," IEEE Trans. Acoust., Speech, and Signal Processing, vol. ASSP-27, pp. 13-18, Feb. 1979.

5. N. C. Gallagher, Jr. and G. L. Wise, "Passband and Stepband Properties of Median Filters," Proceedings of the Princeton Conference on Information Sciences and Systems, March 1980.

6. H. A. David, Order Statistics, (1970), Wiley, New York.

7. F. Kuhlmannand G.L. Wise, "On Spectral Characteristic of Median Filtered Independent Data", Allerton Conference on Communication, Control, and Computing, October 1980.

ON SPECTRAL CHARACTERISTICS OF MEDIAN FILTERED INDEPENDENT DATA

FEDERICO KUHLMANN and GARY L. WISE
Department of Electrical Engineering
University of Texas at Austin
Austin, Texas 78712

ABSTRACT

The bivariate distribution function of a median filtered sequence of independent arbitrary second order random variables is derived. It is then used to qualitatively analyze autocorrelations and spectral densities when the input is discrete valued. The results suggest a low-pass filter characteristic and a low sensitivity to the input distribution. Simulations for various continuous input random variables suggest the same behavior.

I. INTRODUCTION

Linear filtering is not adequate for certain applications. If the signal to be processed consists of the sum of a high frequency noise and a component with information-bearing sharp edges, then a linear filter, upon removing the noise signal will also smear out the edges in the original signal, possibly obscuring their information contents. To overcome this difficulty, nonlinear smoothing devices have been used. Tukey [1] is generally credited with the idea of introducing nonlinear filters based on moving sample medians of the input signal, and he did the pioneering work in this area [2-5]. We will refer to these systems as median filters. Median filters have several interesting characteristics that make them sometimes superior to linear systems. If a signal has impulse-like components, median filtering can eliminate them without significantly modifying other components, and if a signal contains step-like components, median filtering can preserve these discontinuities. Median filters have been studied by several investigators [6-11]; however, many of the reported results are of an empirical nature.

The goal of this paper is the analysis of the output autocorrelation function and the output power spectrum of median filtered stationary sequences of independent data. We present a method for obtaining the output bivariate distribution function, and we use this result to numerically determine the correlation functions for a variety of distributions of discrete valued input signals and for various filter parameters. We then determine the power spectrum of the filter output for these cases. Using simulation results, we estimate power spectra for median filtered continuous amplitude data with several distribution functions. Finally, we note that the results obtained by simulation for continuous input signals are very similar to those obtained by using the bivariate distribution for discrete valued input signals.

II. THE MEDIAN FILTER

Let $\mathscr{X} = \{X_i, i=0,\pm1,\pm2,\ldots\}$ be a sequence of independent identically distributed random variables having the same distribution as the second order random variable X, whose distribution function is given by $F_X(x) = P(X \le x)$. A real number m_X satisfying $P(X \le m_X) \ge 0.5 \le P(X \ge m_X)$ is called a median of X. Consider now a subsequence of $\mathscr{X}$ consisting of an odd number $n = 2N+1$ (for some integer N) of consecutive samples, i.e. $\{X_{j+i}, i=0,\pm1,\ldots,\pm N\}$ for some integer j. If we rearrange these samples in ascending order of magnitude $\{X_j(i), i=1,2,\ldots,2N+1\}$ where $X_j(i) \leq X_j(k)$ if $i < k$, then $X_j(i)$ is the i^{th} order statistic of the subsequence of 2N+1 samples, and in particular $X_j(N+1)$ is the sample median $m(X_{j+i}, i=0,\pm1,\ldots,\pm N)$. It is interesting to notice that if $f(\cdot)$ is a monotonic function, then $m(f(X_{j+i}), i=0,\pm1,\ldots,\pm N) = f(m(X_{j+i}, i=0,\pm1,\ldots,\pm N))$. Special cases of $f(x)$ are $f(x) = x+\alpha$, $f(x) = \alpha x$, and $f(x) = \alpha x + \beta$ where α and β are constant; these allow us to assume zero mean input random variables and/or to change their variances without losing generality. Since digital processing of continuous signals is in fact the processing of quantized versions of the continuous signals, it should be noted that median filtering and monotonic memoryless quantization are commutable operations.

Let us return now to the infinite sequence $\mathscr{X}$. Assume that we slide a window which spans 2N+1 consecutive samples of $\mathscr{X}$ generating a sequence of samples of a new random variable Y; that is, we generate $\mathscr{Y} = \{Y_i, i=0,\pm1,\ldots\}$. If $Y_k = m(X_{k-N}, X_{k-N+1}, \ldots, X_{k+N-1}, X_{k+N})$, then the sliding window will generate sample medians; we will call such a device a median filter with input $\mathscr{X}$, output $\mathscr{Y}$, and parameter N (or window size 2N+1). It is readily seen that the output sequence $\mathscr{Y}$ is strictly stationary. The new random variable Y_1 has a univariate distribution function given by

$$F_{Y_1}(y) = \sum_{j=N+1}^{2N+1} \binom{2N+1}{j} [F_X(y)]^j \, [1-F_X(y)]^{2N+1-j}, \qquad (1)$$

where $\binom{k}{n} = \frac{k!}{n!(k-n)!}$ for $n \leq k$. It is shown in David [12] that if $E[g(X)]$, for some $g(\cdot)$ exists, then $E[g(Y_1)]$ also exists. Thus $\mathscr{Y}$ is a strictly stationary sequence of second order random variables.

Even though the input samples to the filter are independent, the sliding window introduces some dependency among the output samples. In particular, let Y_j and Y_k be generated by the sequences $(X_{j-N}, X_{j-N+1}, \ldots, X_{j+N})$ and $(X_{k-N}, X_{k-N+1}, \ldots, X_{k+N})$ respectively. Assume that $k > j$; if $k-j > 2N$, then $(X_{j-N}, \ldots, X_{j+N})$ and $(X_{k-N}, \ldots, X_{k+N})$ have no samples in common (the window spans do not overlap); therefore Y_k and Y_j are generated by independent samples, which implies the independence of Y_k and Y_j. However, if $k-j \leq 2N$ then the generating sequences have at least one common sample and this in turn implies dependency between Y_k and Y_j. Intuitively it seems reasonable to expect a "large" dependency when the overlap of the generating sequences is large and a "small" dependency when the sequences overlap in just a few samples. In the following we analyze this dependency by means of the output bivariate distribution function.

III. BIVARIATE DISTRIBUTION FOR THE OUTPUT

In this section we will determine the bivariate distribution function for the output of a median filter with parameter N when the input consists of a sequence of independent samples from a random variable X with distribution function $F_X(x) = P(X \leq x)$. From our earlier comments we see that we can assume zero mean without loss of generality. Therefore our analysis will hold for arbitrary second order zero mean input random variables. We are only interested in the region where the sequences generating Y_m and Y_n do overlap. As mentioned earlier, nonoverlapping sequences generate independent outputs and their joint probability distribution can easily be obtained from (1). Let k be the number of samples in which the sequences generating Y_m and Y_n do overlap. Note that $0 \leq k \leq 2N+1$ and assume that $n \geq m$. We now make the following definitions: $\Delta_k = n-m$, $M_1 = \{$samples of the sequence generating Y_m that are not in the overlap$\}$, $M_2 = \{$samples of the sequence generating Y_n that are not in the overlap$\}$, $Q = \{$samples of the overlap$\}$. Since Q contains k elements, M_1 and M_2 have 2N+1-k elements each. In order to consider events with zero probability, we define:

$$\binom{j}{k}^* = \begin{cases} \dfrac{j!}{k!(j-k)!} & \text{if } j \geq k \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

We interpret sums of terms and unions of events indexed by i, from i=j to i=k, when $k < j$ as zero and empty respectively and 0/0 as 0. Based on the input distribution, let

$$S_{k,j}(x) = \binom{j}{k}^* [F_X(x)]^k \, [1-F_X(x)]^{j-k}$$

$$S_{k,j}(x/x \leq y) = \binom{j}{k}^* [F_{X/X \leq y}(x)]^k \, [1-F_{X/X \leq y}(x)]^{j-k}$$

$$S_{k,j}(x/x > y) = \binom{j}{k}^* [F_{X/X > y}(x)]^k \, [1-F_{X/X > y}(x)]^{j-k}$$

where $F_{X/X \leq y}(x) = P(X \leq \min\{x,y\})/P(X \leq y)$ if $P(X \leq y) > 0$ and it is zero otherwise; and $F_{X/X > y}(x) = P(X \leq x, X > y)/P(X > y)$ if $P(X > y) > 0$ and it is zero otherwise. Finally, define the following set of events: $B_{j,k}(x) =$ (in a total of k overlap samples, exactly j of them are less than or equal to x), with corresponding probabilities

$$P(B_{j,k}(x)) = \binom{k}{j}^* [F_X(x)]^j \, [1-F_X(x)]^{k-j} = S_{j,k}(x) \, . \tag{2}$$

At this point we note that the bivariate distribution of Y_m and Y_n is given by

$$P(Y_m \leq y_m, Y_n \leq y_n) = P(\text{at least } N+1 \text{ of the samples generating } Y_m \text{ are } \leq y_m \text{ and at least } N+1 \text{ of the samples generating } Y_n \text{ are } \leq y_n). \tag{3}$$

We will now determine this function using the law of total probability. That is,

$$P(Y_m \leq y_m, Y_n \leq y_n) = \sum_{\mu=0}^{k} P(Y_m \leq y_m, Y_n \leq y_n | B_{\mu,k}(y_n)) P(B_{\mu,k}(y_n)) \tag{4}$$

where in this case μ represents the number of samples in Q that are less than or equal to y_n. Now we analyze the individual terms of the sum in (4). Note that

$P(Y_m \le y_m, Y_n \le y_n | B_{\mu,k}(y_n))$

$$= P(\text{at least } N+1 \text{ samples in } M_1 \cup Q \text{ are less than or equal to } y_m \text{ and at least } N+1 \text{ samples in } M_2 \cup Q \text{ are less than or equal to } y_n | B_{\mu,k}(y_n))$$

$$= P(\text{at least } N+1 \text{ samples in } M_1 \cup Q \text{ are less than or equal to } y_m | B_{\mu,k}(y_n)) \cdot P(\text{at least } N+1-\mu \text{ samples in } M_2 \text{ are less than or equal to } y_n). \qquad (5)$$

Now let $\pi_1(y_m,y_n) = P(\text{at least } N+1 \text{ samples in } M_1 \cup Q \text{ are less than or equal to } y_m | B_{\mu,k}(y_n))$ and $\pi_2(y_n) = P(\text{at least } N+1-\mu \text{ samples in } M_2 \text{ are less than or equal to } y_n)$. It can be seen that

$$\pi_2(y_n) = P\Big(\bigcup_{i=N+1-\mu}^{2N+1-k} B_{i,2N+1-k}(y_n)\Big) = \sum_{j=N+1-\mu}^{2N+1-k} S_{j,2N+1-k}(y_n) \qquad (6)$$

since the events $B_{j,k}(x)$ and $B_{i,k}(x)$ are mutually exclusive for $i \ne j$. In order to determine $\pi_1(y_m,y_n)$ we consider the following two separate cases.

CASE A: Assume $y_m \le y_n$. Then

$\pi_1(y_m,y_n)$

$$= P(\text{at least } N+1 \text{ samples in } M_1 \cup Q \text{ are less than or equal to } y_m | B_{\mu,k}(y_n))$$

$$= \sum_{j=N+1}^{2N+1-(k-\mu)} \sum_{r=0}^{\mu} \binom{2N+1-k}{j-r}^{*} [F_X(y_m)]^{j-r} [1-F_X(y_m)]^{2N+1-k-(j-r)} \cdot \binom{\mu}{r}^{*} [F_{X/X \le y_n}(y_m)]^{r} [1-F_{X/X \le y_n}(y_m)]^{\mu-r}$$

$$= \sum_{j=N+1}^{2N+1-(k-\mu)} \sum_{r=0}^{\mu} S_{j-r,2N+1-k}(y_m) S_{r,\mu}(y_m/y_m \le y_n). \qquad (7)$$

CASE B: Assume $y_m > y_n$. Then

$\pi_1(y_m,y_n)$

$$= P(\text{at least } N+1-\mu \text{ samples in } M_1 \cup Q \text{ are less than or equal to } y_m | B_{\mu,k}(y_n))$$

$$= \sum_{j=N+1-\mu}^{2N+1} \sum_{r=0}^{k-\mu} \binom{2N+1-k}{j-r}^{*} [F_X(y_m)]^{j-r} [1-F_X(y_m)]^{2N+1-k-(j-r)} \cdot \binom{k-\mu}{r}^{*} [F_{X/X > y_n}(y_m)]^{r} [1-F_{X/X > y_n}(y_m)]^{k-\mu-r}$$

$$= \sum_{j=N+1-\mu}^{2N+1} \sum_{r=0}^{k-\mu} S_{j-r,2N+1-k}(y_m) S_{r,k-\mu}(y_m/y_m > y_n). \qquad (8)$$

If we now substitute (2),(5)-(8) into (4) we finally obtain that for $y_m \le y_n$,

$$P(Y_m \leq y_m, Y_n \leq y_n) = \sum_{\mu=0}^{k} S_{\mu,k}(y_n) \sum_{j=N+1-\mu}^{2N+1-k} S_{j,2N+1-k}(y_n) \sum_{j=N+1}^{2N+1-k+\mu} \sum_{r=0}^{\mu}$$

$$\cdot S_{j-r,2N+1-k}(y_m) S_{r,\mu}(y_m / y_m \leq y_n) \tag{9a}$$

and for $y_m > y_n$,

$$P(Y_m \leq y_m, Y_n \leq y_n) = \sum_{\mu=0}^{k} S_{\mu,k}(y_n) \sum_{j=N+1-\mu}^{2N+1-k} S_{j,2N+1-k}(y_n) \sum_{j=N+1-\mu}^{2N+1} \sum_{r=0}^{k-\mu}$$

$$\cdot S_{j-r,2N+1-k}(y_m) S_{r,k-\mu}(y_m / y_m > y_n) \; . \tag{9b}$$

This expression represents the bivariate distribution function for the output samples which are separated by not more than 2N+1 samples. We note that (9) yields one function of two variables for each possible value of the overlap k in the range [1,2N+1].

IV. EXAMPLES

Finding a closed form expression for the output autocorrelation seems to be cumbersome even for small input alphabet sizes. So, in order to obtain some insight into the second-order filter behavior, we use (9) to numerically calculate the output autocorrelation for various finite input alphabets. For filters with parameters 1, 2 and 3, the values of the autocorrelation are presented in Tables I-IV for the indicated input alphabets Z and distributions.

Δ_k	N=1	N=2	N=3
0	1	1	1
1	0.5000	0.6250	0.6875
2	0.2500	0.4375	0.5313
3	0	0.2813	0.4063
4		0.1406	0.2969
5		0	0.1953
6			0.0977
7			0

Table I. Binary input, $Z = \{-1,1\}$, $P(-1) = P(1) = 1/2$

Δ_k	N=1	N=2	N=3
0	0.5097	0.4099	0.3456
1	0.2667	0.2647	0.2422
2	0.1291	0.1842	0.1864
3	0	0.1173	0.1417
4		0.0568	0.1022
5		0	0.0660
6			0.0320
7			0

Table II. Ternary input, $Z = \{-1,0,1\}$, $P(-1) = P(1) = 0.333$

The following two tables present the output autocorrelation when the input consists of standard Gaussian random variables which have been quantized in the minimum mean-squared error sense [13] to the indicated number of levels.

Δ_k	N=1	N=2	N=3
0	0.4738	0.3051	0.2236
1	0.2605	0.2111	0.1701
2	0.1234	0.1494	0.1348
3	0	0.0958	0.1042
4		0.0466	0.0761
5		0	0.0497
6			0.0244
7			0

Table III. Gaussian quantized to 16 levels.

Δ_k	N=1	N=2	N=3
0	0.4615	0.2982	0.2216
1	0.2547	0.2077	0.1701
2	0.1210	0.1475	0.1353
3	0	0.0949	0.1049
4		0.0463	0.0768
5		0	0.0503
6			0.0248
7			0

Table IV. Gaussian quantized to 32 levels.

It is interesting to note that for a given filter parameter, the values of the autocorrelation function normalized with the output variance (correlation at $\Delta_k = 0$) are very similar. To investigate this further, we calculate the normalized power spectral density $\Phi(\omega)$, i.e.,

$$\Phi(\omega) = \sum_{\Delta_k=-\infty}^{\infty} \frac{R(\Delta_k)}{R(0)} e^{-j\Delta_k\omega}$$

for these examples. Figures 1-3 are the plots for different parameters. These plots suggest a low pass characteristic of median filters when the input consists of a sequence of independent samples. These characteristics seem to be relatively insensitive to the input distribution and to the size of the input alphabet. We can also say that the large parameter filters show a narrower "pass band" than the filters with small parameters. All of them, however, display a low pass characteristic.

Finally, for purposes of comparison, we present in Table V simulation based estimates of the power spectra of median filter outputs when the input is a zero mean, unit variance continuous random variable with a Laplace, Gaussian or uniform distribution. The filter parameter is 1 or 2. We note that the filter behavior is similar to that obtained for discrete input signals by using the bivariate distribution approach. The normalized power spectra for the filter outputs obtained by simulating continuous amplitude input sequences are shown in Figure 4 for the above mentioned distributions and filter parameters 1 and 2.

Frequency	Laplace	Gaussian	Uniform
0	2.6312	2.5926	2.6872
π/8	2.3898	2.3599	2.4377
π/4	1.7710	1.7625	1.7982
3π/8	1.0349	1.0491	1.0371
π/2	0.4592	0.4858	0.4416
5π/8	0.2003	0.2237	0.1732
3π/4	0.2290	0.2375	0.2018
7π/8	0.3750	0.3673	0.3520
π	0.4504	0.4358	0.4296

(a) Filter Parameter = 1

Frequency	Laplace	Gaussian	Uniform
0	4.2756	4.4034	4.6804
π/8	3.2168	3.2830	3.4304
π/4	1.2626	1.2356	1.1802
3π/8	0.2611	0.2281	0.1387
π/2	0.2854	0.2938	0.2682
5π/8	0.3239	0.3260	0.2898
3π/4	0.1654	0.1320	0.0610
7π/8	0.1982	0.1629	0.1410
π	0.2976	0.2738	0.3008

(b) Filter Parameter = 2

Table V. Normalized power spectra for continuous inputs.

V. CONCLUSION

We have found an expression for the bivariate distribution function of a median filtered sequence of independent data. This expression was used to numerically analyze correlation functions and power spectra for several input distributions. In the cases that we studied, given a filter parameter N, the power spectrum of the output shows a low pass characteristic and suggests a low sensitivity to the input alphabet size as well as to the input distribution.

As a final comment, the analysis carried out in Section III can easily be reproduced for any other order statistic, yielding the bivariate distribution for a "running order statistic" filter. If we consider specifically the filter output to be the q-th order statistic, $1 \leq q \leq 2N+1$, then substituting q for (N+1) in (3) and reproducing the analysis step by step yields an expression similar to (9) for the bivariate output distribution.

ACKNOWLEDGEMENT

This research was supported by Consejo Nacional de Ciencia y Tecnologia, Mexico; the Department of Defense Joint Services Electronics Program under Contract F49620-77-C-0101; and the Air Force Office of Scientific Research under Grant AFOSR-76-3062.

REFERENCES

1. J. W. Tukey, *Exploratory Data Analysis,* Reading, MA.: Addison-Wesley, 1971 (preliminary edition), see [2,3,4].

2. A. E. Beaton and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," *Technometrics,* vol. 16, pp. 147-185, May 1974.

3. J. W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in *Conf. Rec., 1974 EASCON,* p. 673.

4. J. W. Tukey, *Exploratory Data Analysis,* Reading, MA.: Addison-Wesley, 1977.

5. J. W. Tukey, "The ninther, a technique for low-effect robust (resistant) locations in large samples," in H. A. David, ed., *Contributions to Survey Sampling and Applied Statistics,* Academic Press, New York, 1978.

6. L. R. Rabiner, M. R. Sambur, and C. E. Schmidt, "Applications of a nonlinear smoothing algorithm to speech processing," *IEEE Trans. Acoustics, Speech, Signal Processing,* vol. ASSP-23, pp. 552-557, December 1975.

7. N. S. Jayant, "Average and median-based smoothing techniques for improving digital speech quality in the presence of transmission errors," *IEEE Trans. Communications,* vol. COM-24, pp. 1043-1045, September 1976.

8. T. S. Huang, G. J. Yang, and G. Y. Tang, "A fast two-dimensional median filtering algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-27, pp. 13-17, February 1979.

9. B. R. Frieden, "A new restoring algorithm for the preferential enhancement of edge gradients," *J. Opt. Soc. Amer.,* vol. 66, pp. 280-283, 1976.

10. V. K. Aatre, E. Ataman, and K. M. Wong, "Median filtering," *Proc. of the 17th Annual Allerton Conference on Communication, Control, and Computing,* Monticello, IL., October 10-12, 1979.

11. N. C. Gallagher Jr. and G. L. Wise, "A theoretical analysis of the properties of median filters," *Proc. 1980 Conf. on Information Sciences and Systems,* Princeton University, March 26-28, 1980.

12. H. A. David, *Order Statistics,* John Wiley, New York, 1970.

13. J. Max, "Quantizing for minimum distortion," *IRE Trans. Information Theory,* vol. IT-6, pp. 7-12, January 1960.

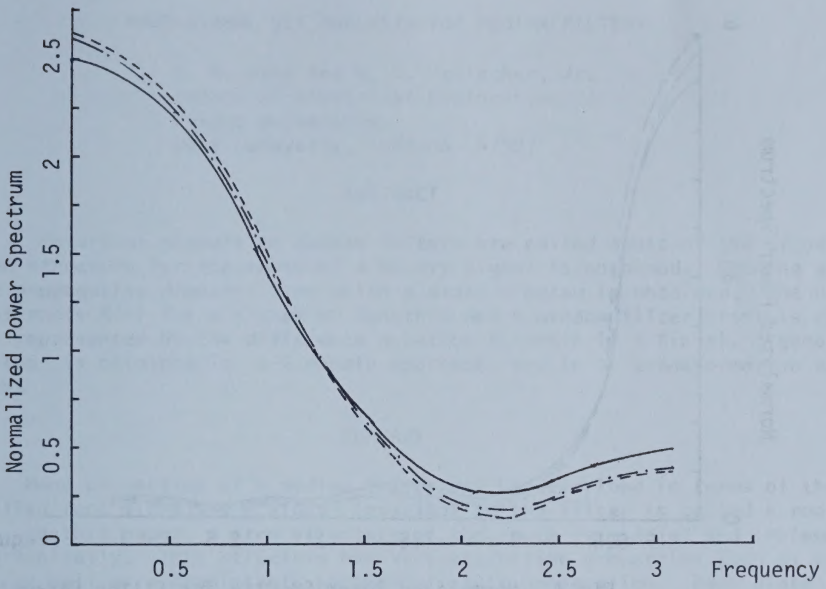


Fig.1 Normalized Power Spectra for Filter Parameter=1

Alphabet Sizes ——— :2
— · — :8
- - - - :16
Quantized Gaussian

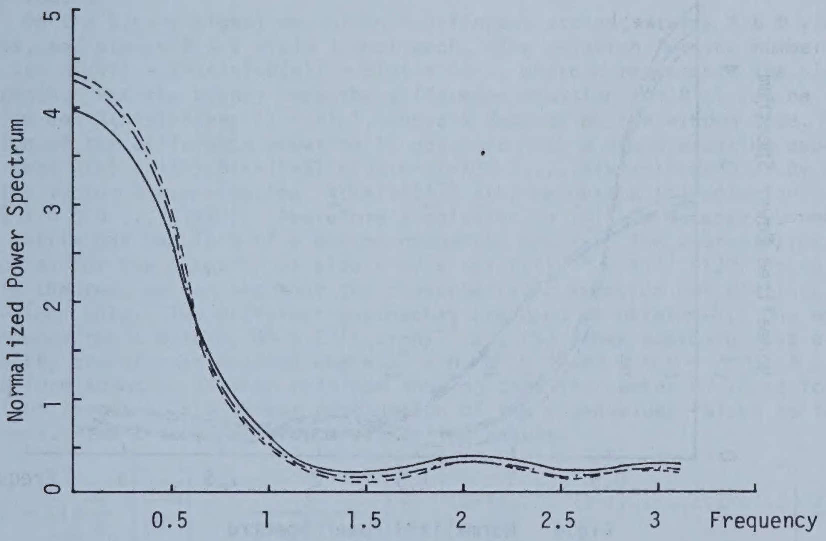


Fig.2 Normalized Power Spectra for Filter Parameter=2

Alphabet Sizes ——— :2
— · — :8
- - - - :16
Quantized Gaussian

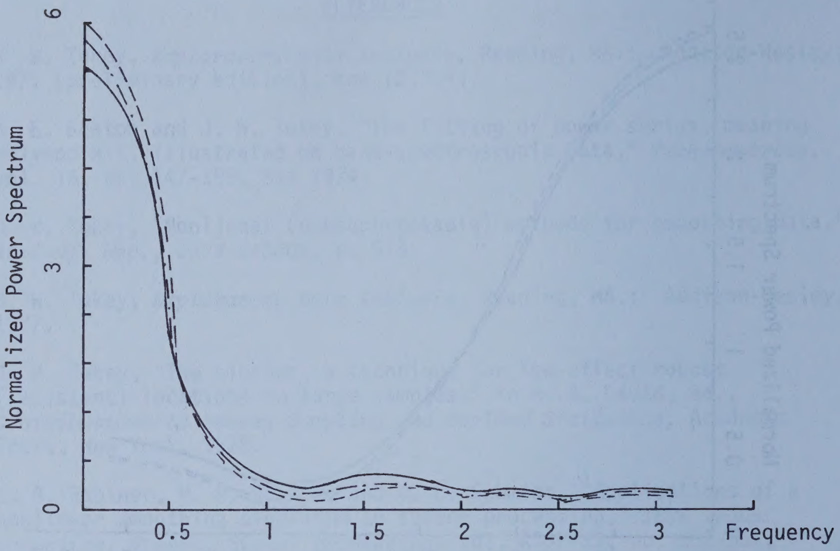


Fig.3 Normalized Power Spectra for Filter Parameter= 3
Alphabet Sizes ——:2
—·—:8 } Quantized Gaussian
----:16

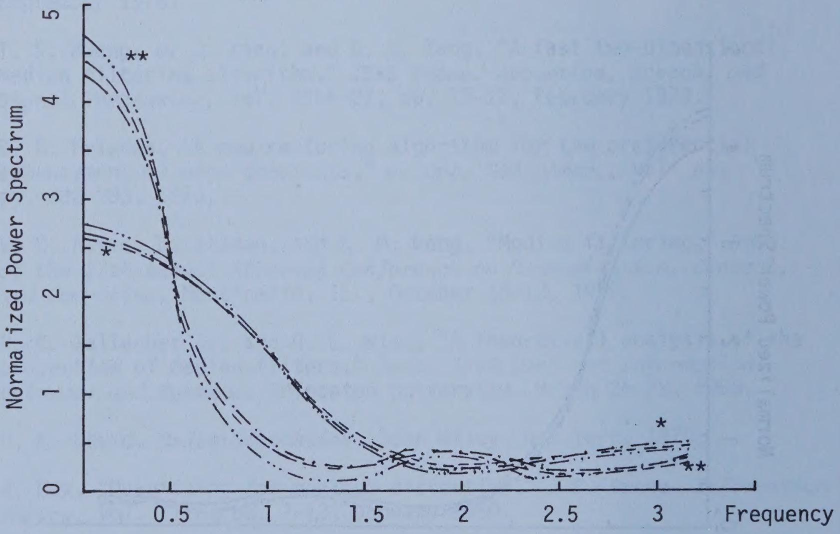


Fig.4 Normalized Power Spectra
Filter Parameters *:1
**:2
Input Distributions —·— :Laplace
---- :Gaussian
—···— :Uniform

ROOT-SIGNAL SET ANALYSIS FOR MEDIAN FILTERS

G. R. Arce and N. C. Gallagher, Jr.
School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907

ABSTRACT

Invariant signals to median filters are called roots of the signal. A tree structure for the roots of a binary signal is obtained. Showing a state propagation property from which a state diagram is obtained. The number of roots R(n) for a signal of length n and a window filter 2*s-1 is exactly represented by the difference equation R(n)=R(n-1) + R(n-s). A general solution is obtained in a Z domain approach, and in a transformation approach.

SUMMARY

Many properties of a median filter may be described in terms of the so called root signals. A signal invariant to the filter is called a root.

In this paper, a tree structure of the roots is modeled and implemented graphically. This structure has very attractive properties such as symmetry as well as a predictable pattern of state propagation. Each state in the tree generates other states, not necessarily of the same kind; then, the new states generate another group of states and so the tree structure follows. The repetition of states in a tree is a function of the length of the signal, and the number of different kinds of states is a function of the filter window size. At each stage along the tree, each state yields a number of roots.

On the binary signal we obtain 4 different states, states A & D yield 2 roots, and states B & C yield 1 root each. The relation for the number of roots is: $R(n+1) = 2*(A(n)+D(n)) + B(n) + C(n)$, where n represents the signal length. For the binary case the difference equation for R(n) can be shown to be: $R(n+s)=R(n+s-1) + R(n)$, where s depends on the window size. The solution of the difference equation is obtained with a state equation approach. Let: $R(k) = X1(k); R(k+1)=X2(k); R(k+2)=X3(k), \ldots, R(k+s-1)=Xs(k)$. By solving the vector state equation $\underline{X}(k+1)=[\underline{A}]^k \underline{x}(k)$ we obtain the solution: $R(k)=[\,1\ 0\ 0\ 0\ \ldots\ 0]\underline{X}(k)$. Therefore a solution to $[\underline{A}]^k$ is necessary, where the $\underline{A}$ matrix has the form of a bottom companion matrix. The characteristic polynomial for the $\underline{A}$ matrix of size s by s is: $f(X)=X^s - X^{s-1} - 1$. Using Sturm's theorem, we can see that the characteristic function has distinct eignevalues only. Two different approaches are used to obtain A^k. One approach used the Z domain, $A^k = Z^{-1}\{(zI-A)^{-1}\ z\}$, the other approach uses a similarity transformation: $X=MQ$ where $\underline{A}^k = \underline{M}\ \underline{D}^k\ \underline{M}^{-1}$ and $\underline{D} = \underline{M}^{-1}\ \underline{A}\ \underline{M}$. A closed form solution is then obtained showing that the number of roots for a signal of length k is a linear combination of the eigenvalues raised to the kth power. The Z domain approach yields the result:

$$R(k_o) = \lim_{z\to\infty} \frac{1}{k_o}\left[\frac{-Zd}{dZ}\right]^{k_o} \frac{1}{Z^s - Z^{s-1} - 1}\left[Z^{s-1}(Z-1), Z^{s-2}(Z-1), \ldots, Z(Z-1), Z\right]\underline{X}(o)$$

where k_o is a specific signal length. In the paper we analyze in detail every point touched in this summary.

The authors gratefully acknowledge the support of the Air Force Office of Scientific Research under grant AFOSR 78-3605.

Nonlinear Regression Model Parameter Estimation Through Random Ordering of Measurements

T. G. Robertazzi and S. C. Schwartz

Dept. of Electrical Engineering and Computer Science
Princeton University
Princeton N.J. 08540

Abstract

An extended Kalman filter is prone to diverge if applied to a nonlinear regression model where minimizing the error over a local portion of the data does not ensure that it is minimized globally. Processing the measurements in a random order, rather than in the causal one in which they occur, can avoid this problem. This is illustrated through simulations involving a particular model.

I: Introduction

Consider a nonlinear regression model in the state variable form:

$$\underline{x}_{k+1}=\underline{x}_k \qquad (1.1)$$

$$y_k=h(\underline{x}_k)+v_k$$

Here $\underline{x}$ is the vector of parameters to be estimated, $h(\underline{x}_k)$ is a known differentiable function of $\underline{x}$ and k, and v_k is an independent Gaussian random variable distributed as $\eta(0,\sigma^2)$ where σ^2 is known. Let N be the number of measurements and let n be the dimension of $\underline{x}$.

A 'sequential estimator' for the above model will be taken to be an estimation algorithm which calculates an updated estimate as each sampled measurement, y_k, arrives. This calculation is made using only the previous estimate, which may take the form of a probability distribution, and the current measurement. Nonlinear sequential estimators, such as the extended Kalman filter, are most successful when an initial reference "trajectory" (initial state/parameter estimates) about which one can linearize is known to a high degree of confidence. If this is not the case and if minimizing the error (between the measurements and the predicted measurement mean) over a local section of the data does not guarantee a good fit globally, then a nonlinear sequential estimator may fail.

An extended Kalman filter, for instance, maintains an error covariance matrix, $\underline{P}_k$, whose trace is non-increasing in the absence of system noise. Practically this means that a good fit during the early part of the waveform can lead to unrealistically low error covariance estimates. These represent such a high degree of assumed confidence in the current parameter estimates that when more recent measurements show a lack of fit, through the growth of the residuals (innovations), the gain is set small enough to ignore them and the filter diverges. Even if the filter gain does not become unrealistically small, parameter estimates which provide a good fit during the latter part of the data may be unreasonable for the early part.

By way of contrast, techniques for linear models, such as the recursive least squares algorithm, do not suffer from this problem since at any time, t_k, the current estimate, $\underline{x}_k$, often can be shown to be the optimal estimate, in some sense, for all measurements up until and including t_k.

II: A Separable Model

The following model will be used for purposes of illustration:

$$h(\underline{x}_k) = (1.0 + \theta_k^2)^{-2.5}(c_1 + c_2\theta_k + c_3\theta_k^2) \tag{2.1}$$

$$\theta_k = s(t_k - t_0)$$

$$s > 0$$

$$\underline{x}_k = [c_1\ c_2\ c_3\ s\ t_0\ ]^T$$

It belongs to the class of 'separable' regression models as some parameters appear linearly and the remaining ones, s and t_0, appear nonlinearly. This class of nonlinear models is significant both because of the wide variety of applications in which it appears and because its structure can be exploited in off-line iterative minimization techniques [1]. Scale and location parameters (s and t_0) often occur in practice, for instance as relative velocity and closest point of approach estimates in tracking problems [3]. What is more relevant is that the separability of (2.1) allows the construction of three dimensional error surfaces as functions of s and t_0 which in an essential way are actually functions of all five parameters.

To do this, the least square estimates of (c_1, c_2, c_3) are computed for various values of (s, t_0) using the noise corrupted data, y_k, of (1.1) and (2.1). For each such set of $(\hat{c}_1, \hat{c}_2, \hat{c}_3, \hat{s}, \hat{t}_0)$ the sum of the square errors between the data and the model is calculated. This error can then be plotted as a function of (s, t_0).

From a theorem appearing in Golub [1] it follows that there is a direct relationship between the extrema in this plot and those in the space of all five parameters. Specifically, a minimum appearing in this type of plot, which is a function of only the nonlinear parameters, is also a minimum in the space of all parameters. Conversely, a minimum appearing in the space of all parameters also appears in this kind of plot as a minimum. This equivalence assumes that the minimum occurs in a small enough neighborhood in the nonlinear parameter space so that the matrix of basis functions has constant rank [13].

Using the model of (1.1) and (2.1), the parameter values of (5.1) and the noisy waveform of Fig. 2, error surfaces using the first 100, 200 and 300 out of 400 equi-spaced measurements were produced. Since minus the $\log_{10}$ of the error is actually plotted, slopes are steeper than they visually appear and a global maximum is desired. As has been indicated, if only 100 measurements are available (Fig. 1) the error surface is quite flat, indicating that many sets of parameter values provide almost equally good fits. Only when most of the measurements are accesible is a maximum apparent. It is interesting to note that these response surfaces are reminiscent of radar ambiguity functions [2] which plot liklihood as a function of unknown parameters (e.g. range and angle) in order to ascertain the uniqueness of maximum likelihood estimates.

Fig. 2 shows the fit produced at the the 165th iteration of an extended Kalman filter which causally processes the noisy data of that figure for the model of (1.1) and (2.1) and which is initialized as in (5.1). It is clear that the fit is satisfactory up until the 165th iteration (measurement). Fig. 3 shows the fit somewhat later, at the 196th iteration. The fit is satisfactory in the immediate neighborhood of the 196th iteration but not elsewhere. Eventually, as Fig. 7 indicates, the covariance matrix estimates become small enough so that the parameter estimates are "frozen" at very inaccurate values.

Several approaches to estimating the parameters of (2.1), either as the measurements arrive or off-line, once they have all arrived, are possible. However the purpose of this paper is to point out the difficulty a sequential estimator such as the extended Kalman filter will encounter in sequentially processing the measurements and to examine the processing of the measurements in a random order by an extended Kalman filter. While this involves storing the measurements off- line, it may still be justifiable in some applications.

III: The RSKF and Gauss-Newton Optimization

The reason that the problem mentioned in section II occurs is that at any iteration the error is only evaluated over a local portion of the data. While it would be preferable to evaluate it over all the data, this can only be done off-line.

Given that all the measurements are available, the simplest way an extended Kalman filter can perform off-line processing is to process the measurements not in their causal order $(y_1, y_2, y_3, \ldots)$ but in a random order, e.g., $(y_{37}, y_{205}, y_{80}, \ldots)$. Over a number of iterations the filter obtains a measure of the error over all the data. Equiprobable sampling without replacement is an obvious procedure for generating a random ordering. An extended Kalman filter which processes the measurements in such an order will be referred to below as a random sampling Kalman filter (RSKF).

The RSKF is closely related to the off-line Gauss-Newton iterative minimization technique (IMT) [8]. Both express the measurement nonlinearity as a first order Taylor series expansion so as to obtain equations that are linear in the state/parameter deviations $(\underline{dx}_k = \underline{x}_{k+1} - \underline{x}_k)$. Linear solution methods can then be applied. In the Gauss-Newton approach N measurement equations are modeled as:

$$\underline{Y} = \underline{\Phi}(\underline{x}) + \underline{\epsilon} \tag{3.1}$$

where each term is a Nx1 column vector and the statistical properties of $\underline{\epsilon}$ are assumed to justify a least squares fit. Expanding the nonlinearity one has:

$$\underline{Y} = \underline{\Phi}(\hat{\underline{x}}_k) + \frac{\partial \Phi(x)}{\partial \underline{x}} \Big|_{\underline{x}=\hat{\underline{x}}_k} \underline{\partial x} + \underline{\epsilon} \tag{3.2}$$

The least squares estimate of the deviation, $\underline{dx}_k = \hat{\underline{x}}_{k+1} - \hat{\underline{x}}_k$, follows from:

$$\min_{\underline{\partial x}} \left\| \underline{Y} - \underline{\Phi}(\hat{\underline{x}}_k) - \frac{\partial \Phi(x)}{\partial \underline{x}} \Big|_{\underline{x}=\hat{\underline{x}}_k} \underline{\partial x} \right\|^2 \tag{3.3}$$

$$\hat{\underline{x}}_{k+1} - \hat{\underline{x}}_k = \hat{\underline{dx}}_k = \left(\frac{\partial \Phi^T(x)}{\partial \underline{x}} \frac{\partial \Phi(x)}{\partial \underline{x}}\right)^{-1} \frac{\partial \Phi^T(x)}{\partial \underline{x}} (\underline{Y} - \underline{\Phi}(\underline{x})) \Big|_{\underline{x}=\hat{\underline{x}}_k} \tag{3.4}$$

In practice (3.4) is multiplied by a scalar which is optimized during each iteration. It is well known that if an extended Kalman filter does not relinearize (i.e., it continues using the original $\underline{x}$ estimate for generating basis functions and partial derivatives), after each measurement is processed, it can be made to produce results identical to those of the single Gauss-Newton iteration above [4]. That is, an extended Kalman filter for (1.1) is a sequential version of the Gauss-Newton iteration where relinearization is performed after each measurement is processed.

The extended Kalman filter equations for the model of (1.1) can be derived from (3.4) in the same way that the recursive least squares algorithm can be derived from the off-line (batch) least squares solution [5]. One obtains:

$$\hat{\underline{x}}_{k+1} = \hat{\underline{x}}_k + \underline{K}_{k,i} [y_i - h_i(\hat{\underline{x}}_k) - \underline{H}_i'(\hat{\underline{x}}_k)(\hat{\underline{x}}_k - \hat{\underline{x}}_0)] \tag{3.5}$$

$$\underline{K}_{k,i} = \underline{P}_k \underline{H}_i'^T(\hat{\underline{x}}_k)[\underline{H}_i'(\hat{\underline{x}}_k)\underline{P}_k \underline{H}_i'^T(\hat{\underline{x}}_k) + \underline{R}_i]^{-1} \tag{3.6}$$

$$\underline{P}_{k+1} = (\underline{I} - \underline{K}_{k,i} \underline{H}_i'(\hat{\underline{x}}_k)) \underline{P}_k \tag{3.7}$$

where $h_i(\underline{x})$ is the measurement function from (2.1) and $\underline{H}_i'(\underline{x})$ is the 1xn vector of partial derivatives of $h_i(\underline{x})$. The subscripts on the gain expression, k and i, represent iteration and randomly chosen measurement, respectively. $\underline{R}_i$ is the measurement noise covariance matrix (a scalar for (1.1)) and $\underline{P}_k$ is the error covarianace matrix.

The above estimator equations correspond to a filter "linearized" about the reference estimate $\underline{x}_0$ [6]. If the equations are relinearized after each measurement $(\hat{\underline{x}}_0 = \hat{\underline{x}}_k)$, the last term in the equation for $\hat{\underline{x}}_{k+1}$ drops out and (3.5) represents the extended Kalman filter equations for (2.1). While such relinearization offers the potential for faster convergence over batch techniques, the convergence point(s) of the extended Kalman filter may be different from that of (batch) iterative minimization techniques such as Gauss-Newton [4,7]. Furthermore, the next section will point out that filter divergence can still be a problem.

IV: Ficticious Measurement Noise

If measurements are causally processed through an extended Kalman filter divergence can occur if many different values of parameter estimates locally provide almost equally good fits. While random sampling can often eliminate this difficulty divergence can still occur for other reasons. Divergence in extended Kalman filters occurs when error sources which are unaccounted for do, in reality, exist so that the error covariance matrix elements take on unrealistically small values [11].

One very simple modification of the filter equations, based on Tenney [3], has been found useful in applying the RSKF to the model of (1.1) and (2.1). A first order approximation of the effect of uncertainties in the parameter estimates upon the measurement equation is:

$$y_i = h_i(\hat{\underline{x}}_k) + \frac{dh_i(\hat{\underline{x}}_k)}{dx_1}\, dx_1 + \frac{dh_i(\hat{\underline{x}}_k)}{dx_2}\, dx_2 + \ldots\ v_i \tag{4.1}$$

This is the measurement model used in both Gauss-Newton optimization and the extended Kalman filter where:

$$\underline{x}_{k+1} - \underline{x}_k = \underline{dx}_k \tag{4.2}$$

and one can linearly solve for $\underline{dx}_k$ and hence $\underline{x}_{k+1}$.

However, proceeding with the philosophy used in Tenney, $\underline{dx}_k$ can be considered to be a zero mean Gaussian random vector whose covariance matrix is the already available estimated error covariance matrix, $\underline{P}_k$. Assuming that this "ficticious measurement noise" is independent of the actual measurement noise, the new measurement noise covariance matrix is:

$$\underline{R}_{k,i}^{new} = \underline{R}_i + \underline{H}_i'(\hat{\underline{x}}_k)\underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k) \tag{4.3}$$

But the second term in the sum is already available in the Kalman gain denominator so that one has:

$$\underline{K}_{k,i} = \underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k)[2\underline{H}_i'(\hat{\underline{x}}_k)\underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k) + \underline{R}_i]^{-1} \tag{4.4}$$

This modification can be generalized. Consider the gain denominator to consist of two measures of uncertainties, one due to the measurement noise and one due to the inaccuracy in the current parameter estimates. The nominal weighting of these two quantities is 1:1. A weighting of 2:1 as indicated in (4.4) corresponds to the statistical model of (4.1). But certainly other weights are possible. That is:

$$\underline{K}_{k,i} = \underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k)[a_k\underline{H}_i'(\hat{\underline{x}}_k)\underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k) + \underline{R}_i]^{-1} \tag{4.5}$$

where a_k is suitably chosen.

As a_k is made greater than one, the rate of filter convergence decreases as the filter behaves as if an increasing amount of uncertainity is present. Specifically, when the RSKF's estimates are far from the optimal estimates, $\underline{H}_i'(\hat{\underline{x}}_k)\underline{P}_k\underline{H}_i'^T(\hat{\underline{x}}_k)$ should be much greater than $\underline{R}_i$ so that the magnitude of the correction vector is $1/a_k$ of what it would be had $a_k=1$.

Tenney's filter implicitly uses $a_k=2$. This does not appear in [3] since the inverse form of the classical Kalman equations (information filter) being used does not contain the term inverted in (4.4). This is also true of other implementations, such as square root filters. However $\underline{R}_i$ can always be replaced during each iteration with the increased $\underline{R}_i^{new}$ of (4.5). Somewhat related, Schmidt's paper [9] on the C-5 navigation system mentions setting $a_k=1.01$ to prevent the covariance matrix from losing positive definiteness due to roundoff error.

Although a weight of $a_k=2$ corresponds to the statistical model of (4.4), it does not necessarily follow that this is the best possible a_k setting. Rather the model of (4.1) is a heuristic device for adding a not necessarily minimal or even sufficient amount of ficticious measurement noise in order to prevent divergence.

The statistically unsatisfactory nature of (4.1) is two fold. First, even though the fictitious noise random variables , $\underline{dx}_k$, and the parameter estimate uncertainty, $\underline{\hat{x}}_k-\underline{x}_k$, are the same (Gaussian) random variables, in going from (4.3) to (4.4) it is assumed that they are uncorrelated. This can be demonstrated more explicity using Schmidt's model for including the effect of parameter uncertainties in the measurement equation [10,12]. Secondly, the effect of parameter uncertainities in the partial derivative evaluations has not been included. The former deficiency would tend to produce a conservative covariance matrix estimate while the latter would have an opposite effect.

V: Simulation Experience

The RSKF was simulated using the classical Kalman filter equations with double precision arithmetic on an IBM 370. One artifice employed to insure satisfactory filter performance was to replace s with e^{β} and estimate β. This replaces the physically constrained s ($s>0$) with the unconstrained β. Otherwise the filter may converge to a local minimum with a negative scale estimate. Note that if the distribution of $\hat{\beta}$ has a standard deviation of δ, the confidence interval $[\hat{\beta}-k\delta,\hat{\beta}+k\delta]$, where k is an arbitrary positive number, corresponds to $[e^{-k\delta}\hat{s},e^{k\delta}\hat{s}]$ in terms of the scale parameter, s.

Figures 4-7 were generated using the following parameter values and estimates:

Table 5.1: Simulation Parameters

Parameter	Actual	Initial	Initial Variance
c_1	0.1	4.0	16.0
c_2	1.0	1.9	1.0
c_3	.15	-4.0	16.0
S	.016	.03	2.4*
T_0	200.0	250.0	1600.0
σ^2	10^{-4}		
N	400		*var. of $\hat{\beta}$

Fig. 4 illustrates the received noisy waveform used for this and the following simulations. The smooth waveform corresponds to the initial parameter estimates.

Fig. 5 is a plot of the relative error of the parameter estimates:

$$\frac{1}{5}\left[\frac{|c_1-\hat{c}_1|}{c_1}+\frac{|c_2-\hat{c}_2|}{c_2}+\frac{|c_3-\hat{c}_3|}{c_3}+\frac{|s-\hat{s}|}{s}+\frac{|t_0-\hat{t}_0|}{t_0}\right] \tag{5.2}$$

Fig. 6 is a plot of the trajectory of the nonlinear parameter estimates, $\hat{s}$ and $\hat{t}_0$. The magnitude of the correction vectors, $\hat{x}_{k+1}-\hat{x}_k$, generally decreases as the filter continues to process measurements. Although the trajectory tends to approach the optimum parameter estimates, it can be seen that successive correction vectors may differ markedly in direction.

Finally Fig. 7 is plot of the covariance matrix trace normalized in the following manner:

$$\frac{1}{5}\left[\frac{P_k(1,1)}{P_0(1,1)}+\frac{P_k(2,2)}{P_0(2,2)}+\frac{P_k(3,3)}{P_0(3,3)}+\frac{P_k(4,4)}{P_0(4,4)}+\frac{P_k(5,5)}{P_0(5,5)}\right] \tag{5.3}$$

The solid curve corresponds to Figs. 4-6 where $a_k=2.0$ while the lower curve corresponds to Fig. 3 where $a_k=1.0$ and the upper curve corresponds to Fig. 2 where the data has been processed causally (and $a_k=1.0$).

Simulations were performed using 50 different random processing orders, the received waveform of Fig. 4 and initial estimates of varying accuracies. The first row of the table below corresponds to the initial estimates of (5.1). The second row represents the same initial estimates except that $\underline{P}_0(4,4)$ = .4 instead of 2.4. The third row is based on even more accurate initial estimates in that $\hat{s}$ = .023 instead of .03 and $\underline{P}_0(4,4)$ = .13. A particular simulation was judged to converge if the final estimates were within 3% of the actual estimates, "close" if they were within 10% and "off" if they were worse than 10%. Recalling the nonlinear transformation used for s, the 95% confidence intervals for s are also listed:

Table 5.4: $a_k = 2.0$

	Convergence	Close	Off
$\underline{x}(4)=.030$ $\underline{P}_0(4,4)=2.4$ $[.045\hat{s}_0, 22.0\hat{s}_0]$	46	2	2
$\underline{x}(4)=.030$ $\underline{P}_0(4,4)=0.4$ $[.28\hat{s}_0, 3.5\hat{s}_0]$	49	-	1
$\underline{x}(4)=.023$ $\underline{P}_0(4,4)=0.13$ $[.50\hat{s}_0, 2.0\hat{s}_0]$	50	-	-

As the rather large uncertainity in $\underline{P}_0(4,4)$ is reduced the filter's performance improves. Simulations using the same initial estimates as in Table 1 and a weight, a_k, of one instead of two were performed and are summarized below. Apparently, without ficticious measurement noise, the filter is so prone to diverge that reducing the uncertainty in $\underline{P}(4,4)$ does not lead to a noticeable improvement.

Table 5.5: $a_k = 1.0$

	Convergence	Close	Off
$\underline{x}(4)=.030$ $\underline{P}_0(4,4)=2.4$ $[.045\hat{s}_0, 22.0\hat{s}_0]$	-	6	44
$\underline{x}(4)=.030$ $\underline{P}_0(4,4)=0.4$ $[.28\hat{s}_0, 3.5\hat{s}_0]$	1	6	43
$\underline{x}(4)=.023$ $\underline{P}_0(4,4)=0.13$ $[.50\hat{s}_0, 2.0\hat{s}_0]$	1	4	45

The square error objective function surface for the model of (2.1) presents inherent practical difficulties for the RSKF or any optimization technique. Two ways in which the RSKF can be misled occur when $s \to \infty$ and when $s \to 0$. In the former case the estimated waveform becomes a narrow spike which ultimately can fit between two adjacent measurement locations so that small changes in the parameter estimates have no effect on the residuals. In the latter case the estimated waveform (in the region being sampled) tends towards a straight line of best fit. Generally the elements of $\underline{P}$ eventually become small enough so that these extreme situations are only partially achieved.

Finally, some nonlinear models can be ill-conditioned to varying degrees with respect to parameter estimation. One example is the model of (2.1) with $(1.0 + \theta_k^2)^{-2.5}$ replaced by $e^{-\theta_k^2}$. If the actual parameters are [.1 .6 .15 .016 180.0], $\sigma^2=10^{-4}$ and the least square estimates are used for the linear (first three) parameters then [.0157 200.0] and [.0156 175] provide equally good fits to three decimal places and [.0161 195.0] is even slightly better. Thus the best estimate of one of the nonlinear parameters can be 10% off from the actual value.

Acknowledgement

The authors would like to acknowledge a number of interesting and useful conversations with Dr. William Schmidt of the Naval Air Development Center, Warminster, Pa. The research reported here was supported in part by the Office of Naval Research under contract N 000-14-80-C-0550.

References

[1] G. H. Golub and V. Pereyra, "The Differentiation of Pseudo-Inverses and Nonlinear Least Squares Problems Whose Variables Separate", SIAM J. of Num. Anal., Vol. 10, No. 2, April 1973, pp. 413-432.

[2] R. A. Altes, "Target Position Estimation in Radar and Sonar, and Generalized Ambiguity Analysis for Maximum Likelihood Parameter Estimation", Proc. of IEEE, Vol. 67, June 1979, pp. 920-930.

[3] R. R. Tenney, R. S. Hebbert and N. R. Sandell Jr., "A Tracking Filter for Maneuvering Sources", IEEE Trans. Autom. Control, April 1977, pp. 246-251.

[4] B. T. Fang, "A Nonlinear Counterexample for Batch and Extended Sequential Estimation Algorithms", IEEE Trans. Autom. Control, Vol. AC-16, No. 1, Feb. 1976, pp. 138-139.

[5] T. C. Hsia, System Identification: Least Squares Methods, D. C. Heath and Co., Lexington, Mass., 1977, pp. 22-25.

[6] A. Gelb, ed., Applied Optimal Estimation, M.I.T. Press, Cambridge, Mass., 1977, 188-189.

[7] L. Ljung, "Asymptotic Behavior of the Extended Kalman Filter as a Parameter Estimator for Linear Systems", IEEE Trans. Autom. Control, Vol. AC-24, No. 1, Feb. 1979, pp. 36-50.

[8] D. W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters, SIAM J., Vol. 11, No. 2, June 1963, pp. 431-441.

[9] S. F. Schmidt, J. D. Weinberg and J. S. Lukesh, "Application of Kalman Filtering to the C-5 Guidance and Control System", in Theory and Applications of Kalman Filtering, C. T. Leondes, ed., NATO Agardograph 139.

[10] S. F. Schmidt, "Application of State-Space Mathods to Navigation Problems", in Advances in Control, C. T. Leondes , ed., Vol. 3, 1966, Academic Press.

[11] B. L. Clark, "A Comparative Evaluation of Several Bearings-Only Tracking Filters", IEEE-ASSP Inter. Conf. 1980, Denver, Colorado, pp. 833-847.

[12] T. G. Robertazzi and S. C. Schwartz, "The Off-Line Use of a Sequential Estimator", Tech. Rep. 4, Information Sciences and Systems Laboratory, Dept. of EECS, Princeton University.

[13] M. R. Osborne, "Some Special Nonlinear Least Squares Problems", SIAM J. of Num. Anal., Vol. 12, No. 4, Sept. 1975, pp. 571-592.

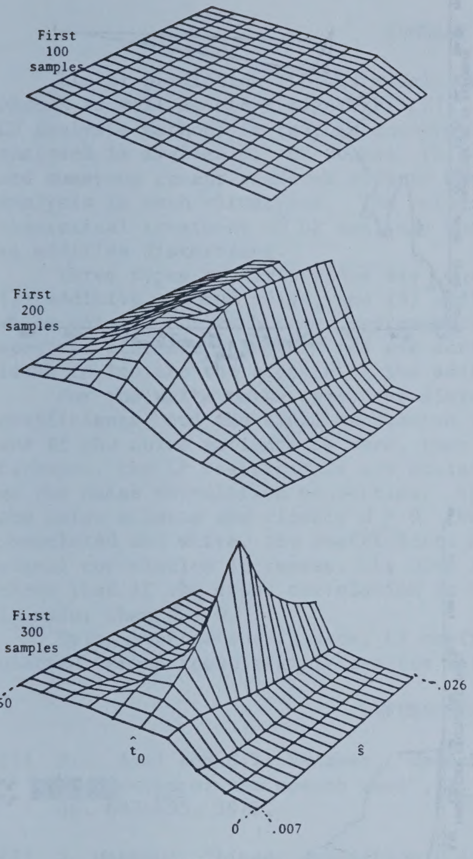


FIGURE 1

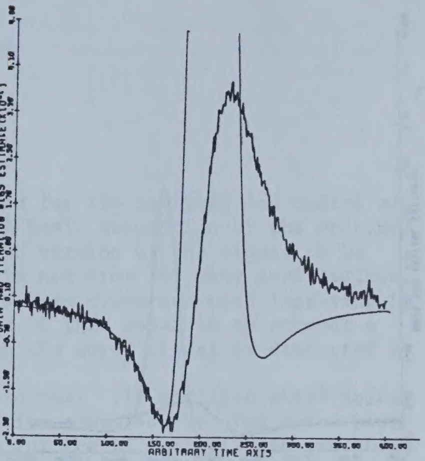


FIGURE 2

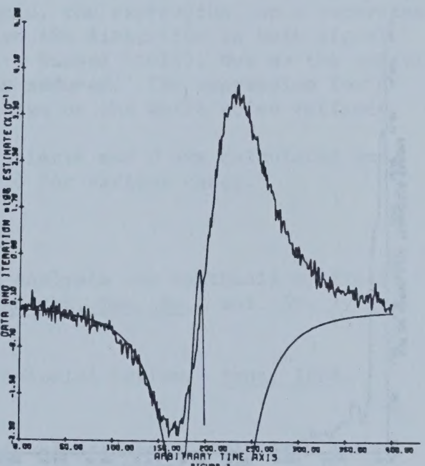


FIGURE 3

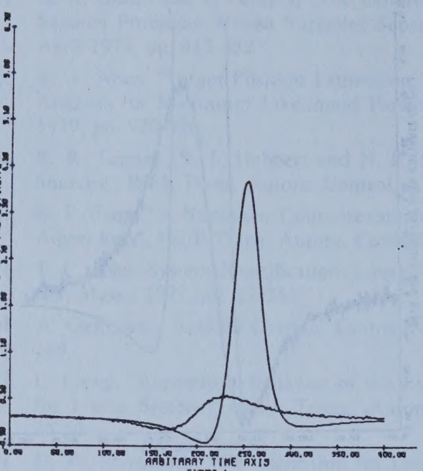


FIGURE 4

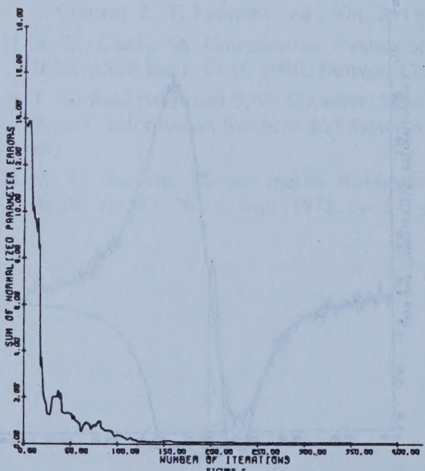


FIGURE 5

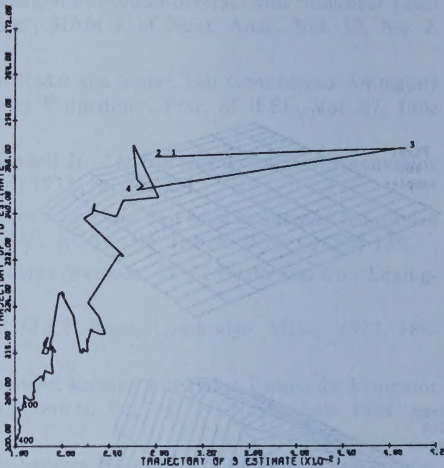


FIGURE 6

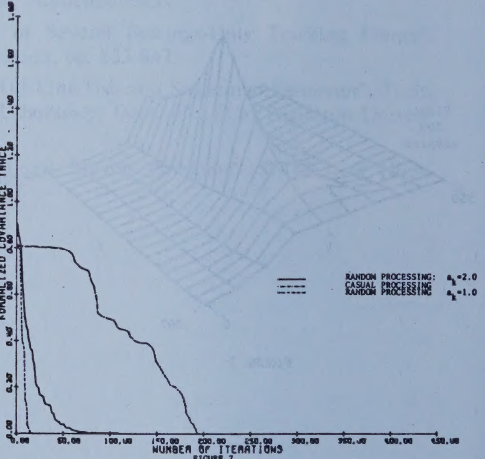


FIGURE 7

THEORY OF LINEAR PREDICTION ANALYSIS FOR DISTORTED INPUTS

JERRY D. GIBSON
Department of Electrical Engineering
Texas A&M University
College Station, Texas 77843

SUMMARY

Linear prediction (LP) is widely used for the analysis and coding of speech as well as other signals [1,2]. A basic assumption of the original LP analysis methods is that an uncorrupted version of the signal to be analyzed is available. Of course, this is not true for many applications, and numerous recent research efforts have been concerned with improving LP analysis in such situations. The purpose of this paper is to present a theoretical treatment of LP analysis when the input signal is distorted by an additive disturbance.

Three types of distortions are considered: (1) additive white noise, (2) additive colored noise, and (3) additive signal-correlated noise plus white noise. For each case, expressions for the LP coefficients and for a spectral distance measure d [3] are derived. The expressions are manipulated to isolate the effects of the additive distortion.

For the white noise case, the distortion produces a bias (only) in the coefficients, and the final expression for d clearly indicates that $d \geq 0$ and if the noise variance is zero, then $d = 0$. For the colored noise disturbance, the LP coefficients are scaled and biased by an amount depending on the noise correlation properties. Again, the expression for d separates the noise effects and clearly $d \geq 0$. When the distortion is both signal correlated and white, the coefficients are biased (only), but as the noise/signal correlation increases, the bias is reduced. The expression for d shows that if the cross correlation is large or the white noise variance is zero, then $d = 0$.

Using average speech data, LP coefficients and d are calculated and plotted versus input signal-to-noise ratio for various cases.

REFERENCES

[1] B.S. Atal and S.L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave", J. Acoust. Soc. Am., vol. 50, pp. 637-655, 1971.

[2] J. Makhoul, "Linear prediction: A tutorial review," Proc. IEEE, vol. 63, pp. 561-580, 1975.

[3] F. Itakura, "Minimum prediction residual principle applied to speech recognition," IEEE Trans. Acoust., Speech, and Signal Processing, vol. ASSP-23, pp. 67-72, 1975.

IMAGE RESTORATION USING RANDOM FIELD MODELS

R. L. KASHYAP* AND R. CHELLAPPA*,†
School of Electrical Engineering
Purdue University
West Lafayette, IN 47907

Abstract

In this paper we consider the use of two classes of two-dimensional Random Field (RF) models known as the simultaneous and conditional Markov RF models to develop algorithms for the minimum mean squared error (MMSE) restoration of degraded images. The degradation is assumed to be due to a space invariant, periodic, non separable point spread function (PSF) and an additive white Gaussian noise. The restoration algorithm is optimal with respect to MMSE criterion if the parameters characterizing the RF models are exactly known. However, in practice the parameters are estimated from the original image. We give an iterative scheme for the estimation of unknown parameters. The restoration algorithm has a general structure that includes various types of RF models, such as causal, semi causal and non causal models. Examples of restoration are given.

Introduction

The restoration of degraded images is of interest in many varied disciplines including space imagery and biomedical images [1-2]. Several approaches have been suggested in the literature using recursive Kalman filter type algorithms [3,4], inverse filtering methods and partial differential equation models [5,6]. The Kalman filter algorithms assume an underlying causal representation for the images, displayed as a state space model. The two dimensional partial differential equations lead to the causal, semi causal and non causal models. By making appropriate assumptions regarding the boundary pixels, the original observations are decomposed into two parts such that the covariance matrix corresponding to one set is diagonalized by sine transforms and the other part corresponds to the boundary response. The important feature is that the eigenfunction expansion in terms of sine function enables fast implementation of the restoration filters.

We consider the use of RF models for image restoration problem. The RF models considered in this paper represent the statistical dependency of a pixel on the neighbors in all directions. Typically, such a dependency could be described by a neighborset consisting of east, west, north and south neighbors or some additional diagonal neighbors. The neighborset may include neighbors other than the nearest neighbors. There are two nonequivalent ways of characterizing the statistical dependency in an image, using the so called simultaneous models [7,8] and conditional Markov models [9,10]. These models are nonequivalent in the sense, that given a simultaneous model an equivalent conditional model always exists, usually having more parameters than the simultaneous models, but the converse is not true. For instance there is no equivalent simultaneous model for the 4-neighbor conditional Markov model in [9].

---

*Supported in part by National Science Foundation under grant NSF-ENG-78-1827.

†Also at Computer Vision Laboratory, Computer Sciences Ctr., University of Maryland, College Park, Maryland 20742. Supported in part by U.S. Air Force Office of Scientific Research, under the grant AFOSR-77-3271.
The continued interest and encouragement of Prof. A. Rosenfeld is sincerely acknowledged, as is the help of Dr. E. J. Delp in providing the image data.

For a finite image the neighbors in all directions are not defined for boundary pixels. This difficulty is overcome in this paper by assuming that the images are represented on a torus lattice [8], [10]. The representation of finite images on torus lattices leads to elegant solutions for the image restoration problems [8]. Since this representation leads directly to covariance matrices that have a block circulant structure, Fast Fourier Transforms (FFT) can be used for the implementation of restoration filters. Further, conditional Markov models where the neighborset dependency extends in all directions has not been paid much attention to in the image restoration literature. Block circulant approximation to block toeplitz covariance structures have been used in the literature [1][11] to enable computation by means of FFT algorithms. But, the block circulant structure that arises in our case is due to the underlying RF model, i.e., the covariance matrices in our paper are exactly block circulant.

In the restoration methods that involve the assumption of an underlying RF model, the model is characterized by a set of parameters. Some arbitrary values for parameters can be assumed or the parameters can be estimated from the original image. (Though for the purposes of this paper we do assume that a noise free original image obeying the underlying RF model is available, restoration methods can be developed as indicated later such that only the noisy image is used.) By imposing a Gaussian structure on the noise driving the RF model, we give an iterative scheme to estimate the parameters.

2. Restoration Schemes Using the Simultaneous Random Field Models.

A: Derivation of the Optimal Filter:

Consider a MXM zero mean, undegraded image represented by a two-dimensional array of gray levels $\{y(s),\ s = (i,j) \in \Omega\}$, $\Omega = \{s = (i,j),\ 1 \le i,\ j \le M\}$ where $y(s)$ is the gray level of the cell s. Since for a finite image, the neighbors for the boundary points are not defined, the image is assumed to be folded into a torus so that (2.1) is valid.

$$y[(i,j) + (i_1,j_1)] = y[(i + i_1 - M) \bmod M + 1,\ (j + j_1 - M) \bmod M + 1] \qquad (2.1)$$

The basic equation of a simultaneous RF model is as in (2.2)

$$y(s) = \sum_{(i,j)\in N} \theta_{i,j}\, y(s + (i,j)) + \sqrt{\rho}\, \omega(s),\ s\in\Omega, \qquad (2.2)$$

In (2.2), $\{\omega(s),\ s\in\Omega\}$ is a sequence of independently distributed random variables of zero mean and unit variance and N denotes the neighbor set, the neighbors on which the observation at $y(s)$ is dependent upon. Equation (2.2) is characterized by a set of parameters $\{\theta_{i,j},\ (i,j)\in N,\ \rho\}$. Denoting $\underset{\sim}{y}$ and $\underset{\sim}{\omega}$ as $M^2 \times 1$ vectors of lexicographic ordered arrays $\{y(.)\}$ and $\{\omega(.)\}$, (2.2) can be written as in (2.3):

$$B(\underset{\sim}{\theta})\, \underset{\sim}{y} = \sqrt{\rho}\, \underset{\sim}{\omega}, \qquad (2.3)$$

where $B(\underset{\sim}{\theta})$ is a $M^2 \times M^2$ block circulant matrix and $\underset{\sim}{\theta}^T = [\theta_{i,j},\ (i,j)\in N]$ Let m be the dimension of $\underset{\sim}{\theta}$.

Assuming that the matrix $B(\underset{\sim}{\theta})$ has an inverse, we have

$$\underset{\sim}{y} = B(\underset{\sim}{\theta})^{-1}\,\underset{\sim}{\omega} \tag{2.4}$$

and the image covariance matrix

$$\underset{\sim}{Q} = E(\underset{\sim}{y}\underset{\sim}{y}^T) = \rho\left[B^T(\underset{\sim}{\theta})\ B(\underset{\sim}{\theta})\right]^{-1} \tag{2.5}$$

is also block circulant.

Consider the case when the image $\underset{\sim}{y}$ is degraded by a non-separable, spatially invariant, periodic PSF represented by a block circulant matrix $\underset{\sim}{H}$ and an additive white noise. Thus if $\underset{\sim}{x}$ and $\underset{\sim}{\eta}$ denote $M^2 x1$ vectors of lexicographic ordered arrays $\{x(.)\}$ and $\{\eta(.)\}$, where $\eta(.)$ is the zero mean signal independent additive noise of variance γ, then

$$\underset{\sim}{x} = \underset{\sim}{H}\underset{\sim}{Y} + \underset{\sim}{\eta} \tag{2.6}$$

we assume for the present that the PSF matrix $\underset{\sim}{H}$ and the parameters $(\underset{\sim}{\theta},\rho)$ in (2.1) are completely known. The restoration problem is posed as follows: determine a $\underset{\sim}{\hat{x}}$ a function of $\underset{\sim}{x}$ such that the mean squared error f is a minimum where

$$f = (\underset{\sim}{y} - \underset{\sim}{\overline{x}})^T\ (\underset{\sim}{y} - \underset{\sim}{\overline{x}}) \tag{2.7}$$

and $\underset{\sim}{\hat{x}}$ is that minimizing value of $\underset{\sim}{\overline{x}}$. The optimal estimate $\underset{\sim}{\hat{x}}$ has the following expression [1]

$$\underset{\sim}{\hat{x}} = \underset{\sim}{Q}\underset{\sim}{H}^T\ (\underset{\sim}{H}\underset{\sim}{Q}\underset{\sim}{H}^T + \gamma\underset{\sim}{I})^{-1}\,\underset{\sim}{x} \tag{2.8}$$

Let $\underset{\sim}{f}_{ij} = \text{col}\,[\underset{\sim}{t}_j,\ \lambda_i\,\underset{\sim}{t}_j,\ \ldots,\ \lambda_i^{M-1}\,\underset{\sim}{t}_j],\ i,j = 1,\ \ldots,\ M$

$\underset{\sim}{t}_j = \text{col}\,[1,\ \lambda_j,\ \lambda_j^2,\ \ldots,\ \lambda_j^{M-1}]$, M-vector

$\lambda_i = \exp\left[\sqrt{-1}\ \lambda_o\ (i-1)\right],\ \lambda_o = 2\,\pi/M.$

Let h_{ij} and μ_{ij}, $1 \le i,\ j \le M$ denote the M^2 eigenvalues of the block circulant matrices $\underset{\sim}{H}$ and $B(\underset{\sim}{\theta})$ respectively. Using the fact that the Fourier vectors $\underset{\sim}{f}_{ij}$ are the eigenvectors of $\underset{\sim}{Q}$ and $\underset{\sim}{H}$ with the corresponding eigenvalues $\rho/||\mu_{ij}||^2$ and h_{ij}, $1 \le i,\ J \le M$, the following expression in (2.9) can be written for $\underset{\sim}{\hat{x}}$:

$$\underset{\sim}{\hat{x}} = \frac{1}{M^2}\sum_{\Omega}\underset{\sim}{f}_{ij}\,(\rho h_{ij}^*/(\rho||h_{ij}||^2 + \gamma\ ||\mu_{ij}||^2))\,\underset{\sim}{f}_{ij}^{*T}\underset{\sim}{x} \tag{2.9}$$

where * denotes the complex conjugate operator.

The computation of $\underset{\sim}{\hat{x}}$ can be done using FFT and the computational complexity is $O(M^2 \log M)$.

B. Parameter Estimation

In the above analysis we have assumed that the parameters $(\underset{\sim}{\theta}, \rho)$ characterizing the RF model are known. In practice, however they are estimated from the original image. One of the popular methods of estimation is the method of Maximum Likelihood (ML), which involves obtaining an expression for the likelihood of observations. We first assume that $\{\omega\,(.)\}$, is distributed Gaussian with zero mean and unit variance.

$$\text{Let,} \quad |\,\mu_{ij}(\underset{\sim}{\theta})\,| \neq 0 \quad \forall i,j = 1, \ldots, M.. \tag{2.10}$$

By using the property of the noise $\{\omega\,(.)\}$, and (2.1), the log likelihood of observations $\ell n\, p(\underset{\sim}{y}/\underset{\sim}{\theta},\rho)$ can be written as

$$\ell n\, p(\underset{\sim}{y}/\underset{\sim}{\theta},\rho) = \ell n \det B(\underset{\sim}{\theta}) - (M^2/2)\,\ell n\, 2\pi\rho - \frac{1}{2\rho} \sum_{s\in\Omega} (y(s) - \underset{\sim}{\theta}^T \underset{\sim}{z}(s))^2 \tag{2.11}$$

$$\text{where,} \quad \underset{\sim}{z}(s) = \text{col}\,[y\,(s + (i,j)),\ (i,j)\ \epsilon N]$$

$$\text{But,}\ \ell n \det B(\underset{\sim}{\theta}) = \sum_{(i,j)\epsilon\Omega} \ell n\,(1 - \underset{\sim}{\theta}^T \underset{\sim}{\psi}_{ij}) \tag{2.12}$$

where

$$\underset{\sim}{\psi}_{ij} = \text{col}[\exp(\sqrt{-1}\,\frac{2\pi}{M}\,((i-1)k + (j-1)\ell)),\ (k,\ell)\,\epsilon N]$$

Substitution of (2.12) into (2.11) gives the desired likelihood function.

Due to the log likelihood function being nonquadratic in $\underset{\sim}{\theta}$, a gradient procedure like Newton Raphson method can be used. Using "off the shelf" algorithms may be computationally expensive. We suggest an iterative scheme given below in Theorem 1 to estimate the parameters:

Theorem 1: Let $\{y(s),\ s\epsilon\Omega\}$ obey the RF model as in (2.1) with a Gaussian assumption for the noise sequence $\{\omega\,(.)\}$. An iterative scheme [12] to compute an estimate $(\bar{\underset{\sim}{\theta}}_{t+1}, \bar{\rho}_{t+1})$ at $(t+1)^{th}$ iteration is given below:

$$\bar{\underset{\sim}{\theta}}_{t+1} = [\underset{\sim}{R} - \frac{1}{\bar{\rho}_t}\underset{\sim}{s}]^{-1}\ (\underset{\sim}{V} - \frac{1}{\bar{\rho}_t}\underset{\sim}{u}), \tag{2.13}$$

$$\bar{\rho}_{t+1} = \frac{1}{M^2} \sum_{s\epsilon\Omega} (y(s) - \bar{\underset{\sim}{\theta}}_{t+1}\ \underset{\sim}{z}(s))^2, \tag{2.14}$$

$$\bar{\underset{\sim}{\theta}}_o = \underset{\sim}{s}^{-1}\ \underset{\sim}{u}, \quad \bar{\rho}_o = \frac{1}{M^2} \sum_{s\epsilon\Omega} (y(s) - \bar{\underset{\sim}{\theta}}_o\ \underset{\sim}{z}(s))^2,$$

$$\underset{\sim}{V} = \sum_{(i,j)\varepsilon\Omega} \underset{\sim}{C}_{ij}, \quad \underset{\sim}{R} = \sum_{(i,j)\varepsilon\Omega} (\underset{\sim}{S}_{ij}\, \underset{\sim}{S}_{ij}^{T} - \underset{\sim}{C}_{ij}\, \underset{\sim}{C}_{ij}^{T}),$$

$$\underset{\sim}{C}_{ij}^{T} = \left[\cos \left[\left(\frac{2\pi}{M}\left((i-1)k + (j-1)\ell\right)\right], (k,l)\ \varepsilon N \right], \text{ m-vector}$$

$$\underset{\sim}{S}_{ij}^{T} = \left[\sin \left[\frac{2\pi}{M}\left((i-1)k + (j-1)\ell\right)\right], (k,l)\ \varepsilon N \right], \text{ m-vector}$$

$$\underset{\sim}{\bar{s}} = \sum_{s\varepsilon\Omega} \underset{\sim}{z}(s)\ \underset{\sim}{z}^{T}(s), \quad \underset{\sim}{u} = \sum_{s\varepsilon\Omega} \underset{\sim}{z}(s)\ y(s)$$

Let $(\underset{\sim}{\bar{\theta}},\bar{\rho})$ denote the final estimates of $(\underset{\sim}{\theta},\rho)$. In the implementation of the restoration filters the estimates $\underset{\sim}{\bar{\theta}}$ and $\bar{\rho}$ obtained from Theorem 1 are used in the place of true parameters.

3. Restoration Schemes Using the Conditional Markov RF Models.

Assume that the observation set $\{y(s), s\varepsilon\Omega\}$ from the original image obeys a conditional Markov RF model, whose representation is as in (3.1):

$$y(s) = \sum_{(i,j)\varepsilon N_1} \theta_{i,j}\ y(s + (i,j)) + \sqrt{\beta}\ V(s) \tag{3.1}$$

where $\{V(s), s\varepsilon\Omega\}$ is a stationary correlated noise sequence with mean zero and unit variance. The basic definition of conditional Markov RF model is in (3.2):

$$E[\ y(s)\ |\ \text{all } y(s_1), s, \varepsilon\Omega, s_1 \neq s] = \sum_{(i,j)\varepsilon N_1} \theta_{i,j}\ y(s + (i,j)) \tag{3.2}$$

Where N_1 is the relevant symmetric neighborset. Note the reflection symmetric property constraining the coefficients, $\theta_{i,j} = \theta_{-i,-j}$. (3.1) can be rewritten as,

$$A(\underset{\sim}{\theta})\ \underset{\sim}{y} = \sqrt{\beta}\ \underset{\sim}{V} \tag{3.3}$$

where $A(\underset{\sim}{\theta})$ is a block circulant as well as symmetric matrix. Using (2.10) and (3.3)

$$\underset{\sim}{y} = A^{-1}(\underset{\sim}{\theta})\ \underset{\sim}{V} \tag{3.4}$$

$$\text{The image covariance matrix } \underset{\sim}{Q} = E(\underset{\sim}{y}\underset{\sim}{y}^{T}) = A^{-1}(\underset{\sim}{\theta})\beta \tag{3.5}$$

Assume that the original image y(.) obeys (3.1) and we are given a degraded image related to y(.) by (2.6). Proceeding along similar lines of section 2, it can be shown that the optimal estimate $\underset{\sim}{x}'$ for this case is given as in (3.6):

$$\underset{\sim}{x}' = \frac{1}{M^{2}} \sum_{(i,j)\varepsilon\Omega} \underset{\sim}{f}_{ij}\ \beta h^{*}_{ij}\ /(\beta||h_{ij}||^{2} + \gamma\mu'_{ij}))\underset{\sim}{f}^{*T}_{ij}\underset{\sim}{x}, \tag{3.6}$$

where μ'_{ij}, $1 \leq i, j \leq M$ are the eigenvalues of $A(\underset{\sim}{\theta})$

$$\mu'_{ij} = (1 - 2\underset{\sim}{\theta}^T \underset{\sim}{\phi}_{ij}), \tag{3.7}$$

$$\underset{\sim}{\phi}_{ij} = [\ \cos \frac{2\pi}{M} [(i-1)k + (j-1)\ell],\ (k,l)\,\epsilon N'_1]$$

where N'_1 includes only those members of N_1 that do not possess the reflection symmetric property. Again, FFT computations can be used. Thus far in the analysis, we have assumed that the parameters characterizing the model are known but in practice they are estimated from the original image. An estimation scheme similar to that for simultaneous models has been developed for conditional models.

4. Examples of Restoration

The restoration schemes developed in earlier sections is illustrated by examples. The original picture (256 x 256) is shown in Fig. 1 and Fig. 1b is the noisy picture, with Signal to Noise Ratio (SNR) 7db, SNR being defined as the ratio of signal variance to noise variance. As it is not appropriate to assume that a 256 x 256 image is stationary, we divide the picture into 16 blocks, each block being of size 64 x 64. For each block, the parameters of the model are computed using the iterative scheme of section 2 and N = {(0,1), (0,-1), (-1,0), (1,0)}. Using (2.10) with the estimates replacing the true parameters the noisy picture is filtered block by block. Due to blocking, artificial lines are present in the restored image. These edges are removed by a simple interpolation scheme since the precise locations are known [2]. The final restored picture is shown in Fig. 1C. Fig. 2 is an example of restoration of an image (64 x 64) degraded by Gaussian blur

$$h(k,\ell) = \frac{1}{2\pi} \exp\{-\frac{1}{2}(k^2 + \ell^2)\}$$

and additive white noise of SNR = 5. Several restored images are given.

5. Discussions

We have developed restoration schemes using the simultaneous and conditional Markov RF models defined on a torus lattice. The use of torus lattice representation leads to the use of fast Fourier transforms in the implementation of the filters, for any arbitrary neighborhood. Such fast filters for restoration have been given in the literature, [5,6], for semi causal and non causal models by manipulating the boundary conditions. The decomposition schemes suggested in [5,6] are valid only for isotropic neighborsets. However, our experiments with different models and the estimation method in section 2 do not support this isotropy assumption. Fast Fourier transforms for restoration of images using Wiener filters [1], [11] have been considered earlier by approximating the block Toeplitz covariance structures by block circulant matrices. The important difference between the scheme developed here and the earlier schemes [1][11] is in the characterization of signal statistic. The MMSE schemes need exact knowledge of the two-dimensional Spectral Density Function (SDF). In practice it is not known. Hence, the true SDF is replaced by an estimate. Typical estimates of SDF can be obtained by making suitable assumptions regarding the ACF as in [11] or by using estimates such as FFT of empirical covariance matrices. The former scheme assumes very simple functions for the ACF, while the latter yields inconsistent estimates of the SDF. In our approach, the true two dimensional SDF is an explicit function of the parameters given by [8],

$S_y(\underset{\sim}{\theta},\rho,\gamma_1,\gamma_2) = \rho/||\mu_{ij}(\underset{\sim}{\theta})||^2$ in the case of simultaneous models and $\beta/\mu'_{ij}(\underset{\sim}{\theta})$ in the case of conditional models, where $\gamma_1 = \lambda_o i$ and $\gamma_2 = \lambda_o j$.

By using consistent estimates like $S_y(\overline{\underset{\sim}{\theta}},\overline{\rho},\gamma_1,\gamma_2) = \overline{\rho}/||\mu_{ij}(\overline{\underset{\sim}{\theta}})||^2$ as done here, better performance can be achieved.

6. Extensions

In the development of the restoration scheme, we have assumed that an undegraded noise free original is available and that parameters can be estimated. In practical situations only the noisy image is available. Assume that the degradation is due to additive noise alone of known variance γ. The covariance matrix $\underset{\sim}{Q}'$ of $\{x(.)\}$ is related to the covariance matrix $\underset{\sim}{Q}$ of $\{y(.)\}$ as in (6.1):

$$\underset{\sim}{Q}' = \underset{\sim}{Q} + \gamma I \qquad (6.1)$$

Using (6.1), for the case of degradation due to additive noise alone, (2.8) can be written as

$$\hat{x} = \underset{\sim}{Q}(\underset{\sim}{Q} + \gamma I)^{-1}\underset{\sim}{x} = (\underset{\sim}{Q}' - \gamma I)\,(\underset{\sim}{Q}')^{-1}\underset{\sim}{x} \qquad (6.2)$$

By assuming that the noisy image $\{x(.)\}$ obeys a RF model as in (2.1) characterized by parameters $(\underset{\sim}{\theta}',\rho')$, the eigenvalues of $\underset{\sim}{Q}'$ can be written

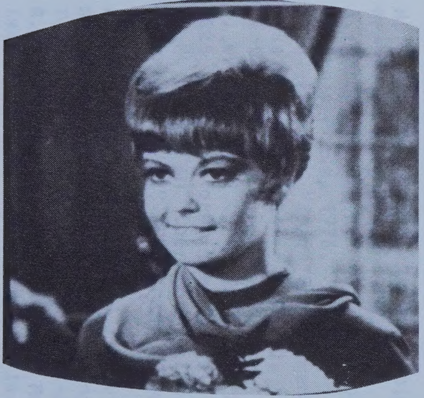
as $\rho'/||\overline{\mu}_{ij}(\underset{\sim}{\theta}')||^2$, $1 \leq i,j \leq M$ where $\overline{\mu}_{ij}(\underset{\sim}{\theta}) = (1 - \underset{\sim}{\theta}'^T \underset{\sim}{\psi}_{ij})$

$$\text{Hence } \underset{\sim}{\hat{x}} = \sum_{\Omega} \underset{\sim}{f}_{ij} \, \frac{(\rho' - \gamma\,||\overline{\mu}_{ij}(\underset{\sim}{\theta}')||^2)}{\rho'} \, \underset{\sim}{f}_{ij}^{*T} \, \underset{\sim}{x}$$

References

1. H. C. Andrews and B. R. Hunt, Digital Image Restoration, New Jersey: Prentice-Hall, Inc., 1977.
2. A. Rosenfeld and A. C. Kak, Digital Image Processing, New York: Academic Press, 1976.
3. B. R. Freiden, "Image Enhancement and Restoration," in Picture Processing and Digital Filtering, (Ed. T. S. Huang). New York: Springer Verlag, 1979.
4. J. W. Woods and C. H. Radewan, "Kalman Filtering in Two Dimensions," IEEE Trans. Inform. Theory, Vol. IT-23, pp. 473-482, July 1977.
5. A. K. Jain, "An Operator Factorization Method for Restoration of Blurred Images," IEEE Trans. Computers, Vol. C-26, pp. 1061-1072, Nov. 1977.
6. A. K. Jain and J. R. Jain, "Partial Difference Equations and Finite Differences in Image Processing - Part II: Image Restoration," IEEE Trans. Automat. Contr., Vol. AC-23, pp. 817-833, Oct. 1978.
7. R. L. Kashyap, "Univariate and Multi-variate Random Field Models for Images," Comput. Graphics and Image Proc., Vol. 12, pp. 257-270, March 1980.

8. R. L. Kashyap, "Random Field Models on Torus Lattices for Finite Images," (submitted).
9. J. W. Woods, "Two-dimensional discrete Markov Random Fields," IEEE Trans. Inform. Theory, Vol. 18, pp. 232-240, Mar. 1972.
10. M. Hassner and J. Sklansky, "The use of Markov Random Fields as Models of Textures," Computer Graphics and Image Proc., Vol. 12, pp., April 1980.
11. W. K. Pratt and F. Davarian, "Fast Computational Techniques for Pseudo Inverse Trans. Computers, Vol. C-26, pp. 571-580.
12. R. Chellappa, "Fitting Random Field Models to Images," Tech. Report, TR-928, Computer Sciences Ctr., Univ. of Maryland, College Pk., 20742, Aug. 1980.



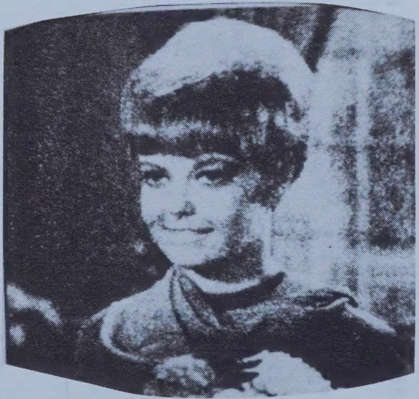
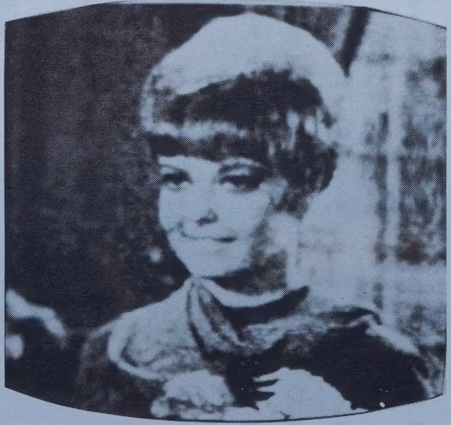


Fig. 1. Restoration of a noisy image. Top: Original, Top Right: Noisy image, (SNR = 7 db), Right: Restored image with N = {(0,1), (0,-1), (-1,0), (1,0)} and simultaneous RF model .



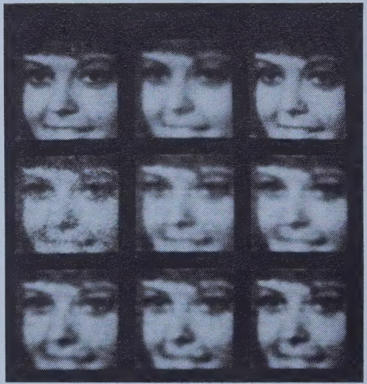
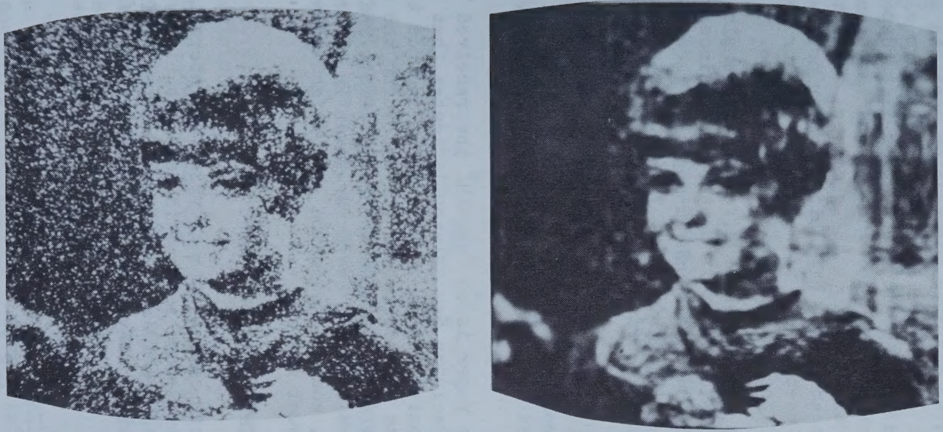


Fig. 2. Restoration of noisy Gaussian blurred image. (1,1): Original, (1,2): Blurred image (1,3): Restored image SNR = ∞, (2,1): Blur + noise (7 db), (2,2): Restoration with N = {(0,-1)(-1,0),(-1,-1)}, (2,3): N = {(-1,0)(1,0),(0,-1)}, (3,1): N = {(0,1),(0,-1),(-1,0),(1,0), (3,2): N = {(0,1),(1,0),(0,-1),(-1,0), (1,-1),(-1,1)}, (3,3): N = {(0,1), (0,-1),(-1,0),(1,0),(-1,1),(1,-1), (-1,-1),(1,1)}



a

b

Fig. 3. Restoration of a Noisy Image. a) Noisy Image, SNR = 0 db, b) Restored Image Using Simultaneous Model and N = $\{(0,1), (0,-1), (-1,0), (1,0)\}$

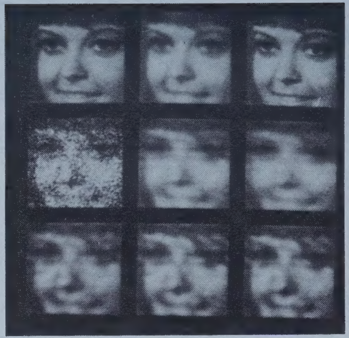


Fig. 4. Similar to Fig. 2, with SNR = 0 db.

PTIMAL MEMORYLESS RECONSTRUCTION OF TRANSFORMED IMAGES TRANSMITTED OVER NOISY CHANNELS

JERRY D. GIBSON, RANDALL C. REININGER, AND JOHN W. MCVAY
Department of Electrical Engineering
Texas A & M University
College Station, Texas 77843

ABSTRACT

Channel error effects in a transform image compression system are reduced without error correcting codes by the use of soft decision demodulation and optimal filtering. Filtering is accomplished only if a received word is unreliable. An optimal reconstruction method is derived and examples are presented.

I. INTRODUCTION

Transform coding is a powerful and popular method for the data compression of images. Transform coding for data compression consists of two operations [1]: First, the set of statistically dependent picture elements (pels) are linearly transformed to a set of independent or "less dependent" coefficients. Second, the coefficients are individually quantized and coded (some may be discarded) for transmission to the receiver. This technique produces good quality images at a rate of 1 bit/pel over noise-free channels. For a probability of (independent) bit error of 10^{-3} or greater however, noticeable degradation of the reconstructed image occurs.

One approach for reducing the channel error effects is to employ a forward error-correcting code. This approach is unattractive since the transmitted data rate is increased and since such an approach requires a modification at the transmitter. This latter point is important if the transmitter is aboard a limited payload craft such as a satellite or a remotely piloted vehicle.

Another approach is to smooth or filter the reconstructed image or the received coefficients to de-emphasize transmission error effects. Unfortunately, if this method is used when there are no errors present, the image is usually degraded.

The approach proposed in this paper is to "smooth" the received transform coefficients only if a soft decision demodulator indicates that the coefficient is unreliable. The contributions of this paper include the derivation of an optimal memoryless transformation for smoothing the coefficients, the inclusion of correlation between the coefficients and the quantization noise, a method for calculating the error power in the coefficients due to the noisy channel, and the inclusion of a weighting matrix to allow modeling of the human visual system.

II. SOFT DECISION DEMODULATION

A block diagram of the complete communication system of interest here is shown in Fig. 1. The modulation method assumed is coherent binary phase shift keying (BPSK) and the channel is assumed to insert additive white Gaussian noise to the transmitted waveform. The probability of an independent bit error is thus

$$P_e = Q(\sqrt{2E/N_0}) = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{2E/N_0}}^{\infty} e^{-t^2/2} dt \qquad (1)$$

where $N_0/2$ is the double-sided noise spectral density, E is the average signal energy, and Q(·) is the Q-function from [2].

The concept of soft decision demodulation used here follows Sundberg [3] and is illustrated in Fig. 2. If a demodulated pulse is outside of the erasure zone, $\pm\sqrt{E}\,T$, the bit is considered reliable, and no smoothing is performed. If the pulse falls within the erasure zone, however, a smoothing transformation is performed on the coefficients. Only the M most significant bits of selected coefficients are monitored by the soft decision logic to minimize complexity. The mean-squared error in a particular coefficient due to channel errors when soft decision demodulation is used is given by [3]

$$\epsilon^2 \cong \left(\sum_{i=1}^{M} A_i P_{D_i} + \sum_{i=M+1}^{N} A_i P_e + \Delta\sigma_s^2 P_R\right)\sigma_s^2 \tag{2}$$

where A_i denotes the ith A-factor, P_D is the probability of an undetected bit error, P_e is the hard decision bit error probability, P_R is the probability that an N-bit word is declared unreliable, and $\Delta\sigma_s^2$ is the mean-squared error in the smoothed value.

The remaining probabilities in Eq. (2) are

$$P_{D_i} = Q\left(\sqrt{\frac{2E}{N_0}}\,[1 + T_i]\right) \tag{3}$$

and

$$P_R = 1 - \prod_{i=1}^{M} (1 - P_{\Delta T_i}) \cong \sum_{i=1}^{M} P_{\Delta T_i} \tag{4}$$

where T_i is the normalized threshold for the ith bit and $P_{\Delta T_i}$ is the probability that the received voltage is in the "erasure" zone,

$$P_{\Delta T_i} = Q\left(\sqrt{\frac{2E}{N_0}}\,[1 - T_i]\right) - Q\left(\sqrt{\frac{2E}{N_0}}\,[1 + T_i]\right) \tag{5}$$

Choosing the thresholds T_i, i = 1,2,...,M, to minimize Eq. (2) yields

$$T_i = \frac{N_0}{4E}\,\log_e\left[\frac{A_i}{\Delta\sigma_s^2} - 1\right] \tag{6}$$

The actual receiver threshold voltage is set at $\sqrt{E}\,T_i$. Since the threshold must be positive, an additional condition that must be met is that

$$A_i > 2\Delta\sigma_s^2 \tag{7}$$

for i = 1,2,...,M, which determines the maximum number of bits usable for soft decision.

The A_i in Eq. (2) are called A-factors, and they indicate the mean-squared error in the coefficient contributed by an error in bit i. In general, the A-factors depend on the number of PCM bits, the signal level, the signal probability density function, the quantization scheme, and the code assignment. As a result, the A-factors for this application are different from those in [3]. In fact, the A-factors will be different depending on which transform is used. For the discrete cosing transform (DCT), the transform coefficients, except for the dc term, can be assumed to be

Gaussian, and the dc component has a uniform probability density function (pdf). Thus, quantizers can be chosen to be matched in the mean squared error sense to a uniform pdf or a Gaussian pdf.

Before calculating A-factors, it is necessary to select a code assignment for the quantizer output levels. Excluding the use of "entropy" coding for the present work, two possible code assignments are the natural binary code (NBC) and the folded binary code (FBC), illustrated in Fig. 3 for N = 3. Another possible coding scheme, called the minimum distance code (MDC), has been studied by Rydbeck and Sundberg [4] for compandored speech signals. The MDC has the property that single bit errors in the most probable code words produce the smallest reconstruction error, and that the reconstruction error due to transmission errors is larger for less probable code words. The MDC is most useful in those systems where the dynamic range can be improved.

To compare the NBC, FBC, and the MDC for the particular application of interest, single bit error A-factors for each code were calculated for an 8-bit Max [5] quantizer with a zero mean, unit variance Gaussian input. These results are presented in Table I. Inspection of these results reveals that the MDC A-factors are more equal in significance than the A-factors for the FBC and the NBC. However, the sum of the A-factors, which represents the total noise power due to single bit errors, is about equal for the FBC and the MDC, and each are much less than the NBC. From these tables it is evident that: (1) The NBC is clearly worse than the other two codes, and (2) the FBC and MDC have equivalent performance when averaged over all single bit errors. As a result, the FBC is selected for use in the remainder of this work, primarily on the basis of ease of implementation.

As an example of the performance improvement to be expected using soft decision demodulation, consider the plot of SNR_{out} versus E/N_0 shown in Fig. 4, where SNR_{out} is defined by

$$SNR_{out} = \frac{\sigma_s^2}{E_q^2 + E_c^2 + E_a^2} = \frac{1}{(E_q^2 + E_c^2)_n + E_{an}^2} \tag{8}$$

In Eq. (8), E_q^2 is the quantization noise power, E_c^2 is the clipping noise power, and E_a^2 is given by Eq. (2). The subscript "n" indicates normalization by σ_s^2. The term $(E_q^2 + E_c^2)_n$ is readily available from tables [5] and E_{an}^2 can be calculated using Eq. (2). To plot Fig. 4, an 8-bit FBC was assumed and the thresholds in Eq. (6) were optimized for $E/N_0 = 4.2$dB, which corresponds to $P_e \doteq 10^{-2}$. Using Table I, M is selected to be 3 for an assumed $\Delta\sigma_s^2 = 0.1$. Table II lists the soft decision thresholds (normalized by $\sqrt{E}$) for the FBC, N = 8, and M = 3. At the design value of $E/N_0 = 4.2$dB, the gain in SNR_{out} from soft decision is 4.77dB, which corresponds to a 1.40dB gain in channel SNR. There is a crossover point at $E/N_0 = 7.60$dB where soft decision becomes inferior to hard decision at higher values of E/N_0. This is because the high thresholds for rejection were selected for $E/N_0 = 4.2$dB, and at large E/N_0 (small P_e), the estimate used for replacement has more error than the received word.

The A-factors for an 8-bit MDC were used to generate a plot similar to Fig. 4. Since the MDC A-factors are more equal in significance, a large M = 7 can be used in soft decision. However, at the E/N_0 design value, the gain in SNR_{out} from soft decision was only 3.32dB for the MDC.

An improvement in SNR_{out} of about 5dB certainly seems attractive, but it is important to note that this result assumes that $\Delta\sigma_s^2 = 0.1$. That is, it was assumed that the mean squared error in the smoothed value of the received signal is 10% of the mean squared value of the actual signal. The validity of this assumption depends on the method used to generate smoothed values to replace unreliable words. One method for generating these smoothed values is developed in the next section.

III. OPTIMAL MEMORYLESS RECONSTRUCTION

If it is decided that a significant bit of a received coefficient is unreliable, then this coefficient must be replaced by a smoothed estimate of some kind. There are many possible approaches to obtaining such estimates. The approach investigated here does not use coefficients from other blocks, but rather, it exploits any dependencies remaining in the current set of transform coefficients as well as any a priori information available concerning the quantization and channel noise. The reconstruction is thus called memoryless.

The general approach followed here was originally proposed by Pratt [6], but he did not include the possibility of signal and noise correlation, he gave no method for computing the required noise variances, and he did not investigate in detail the additive noise case. Consider an nxn data array $\underline{B}$ that is rewritten as an n^2xl column vector given by

$$\underline{b}^T = [b_{11}, b_{12}, \dots, b_{n,n-1}, b_{n,n}] \tag{9}$$

The operations to be performed on this data vector are illustrated in Fig. 5. The input data vector is operated on by a unitary transformation, such as the Hadamard, Slant, Cosine, or Karhunen-Loeve transform [1,7] denoted by $\underline{C}$, to yield

$$\tilde{\underline{b}} = \underline{C}\,\underline{b} \tag{10}$$

The selection matrix $\underline{S}$ with elements s_{ij}, $j = 1,\dots,n$, and $i = 1,\dots,m$, $m < n$, has $s_{ij} = 1$ if the jth component of $\tilde{\underline{b}}$ is to be the ith component of $\tilde{\underline{b}}_T$. The selection matrix would typically retain the m highest energy components and discard the remaining n-m in order to improve data compression. Thus,

$$\tilde{\underline{b}}_T = \underline{S}\,\tilde{\underline{b}} \tag{11}$$

After the channel effects introduce noise into the transmitted coefficients, the received vector is multiplied by a nxm reconstruction matrix $\underline{W}$, which produces an estimate of all spectral components that were present in $\tilde{\underline{b}}$. Hence,

$$\hat{\tilde{\underline{b}}} = \underline{W}\,[\tilde{\underline{b}}_T + \underline{V}] \tag{12}$$

Finally, the inverse unitary transformation is applied to yield a compressed version of the original image,

$$\underline{\hat{b}} = \underline{C}^{-1} \underline{\hat{\tilde{b}}} \tag{13}$$

The focus of this section is on the choice of $\underline{W}$ to optimize overall system performance. Pratt [6] chose $\underline{W}$ to minimize the mean squared error (MSE) between the input signal vector and the receiver output. For the present work, $\underline{W}$ is chosen to minimize the weighted mean squared error (WMSE) given by

$$\varepsilon^2 = \mathrm{tr}\,\{E\,[\underline{R}^{-1}\,(\underline{b} - \underline{\hat{b}})(\underline{b} - \underline{\hat{b}})^T]\}$$

where $\underline{R}^{-1}$ is assumed known and is an as yet unspecified weighting matrix that is chosen based on perceptual experiments. Now, from Eqs. (10)-(13),

$$\begin{aligned}
\underline{\hat{b}} &= \underline{C}^{-1}\,\underline{W}\,[\underline{\tilde{b}}_T + \underline{V}] \\
&= \underline{C}^{-1}\,\underline{W}\,[\underline{S}\,\underline{\tilde{b}} + \underline{V}] \\
&= \underline{C}^{-1}\,\underline{W}\,[\underline{S}\,\underline{C}\,\underline{b} + \underline{V}]
\end{aligned} \tag{15}$$

which upon substitution into Eq. (8) yields

$$\begin{aligned}
\varepsilon^2 = \mathrm{tr}\,\{E\,[\underline{R}^{-1}\,&(\underline{b} - \underline{C}^{-1}\,\underline{W}\,\underline{S}\,\underline{C}\,\underline{b} - \underline{C}^{-1}\,\underline{W}\,\underline{V}) \cdot \\
&(\underline{b} - \underline{C}^{-1}\,\underline{W}\,\underline{S}\,\underline{C}\,\underline{b} - \underline{C}^{-1}\,\underline{W}\,\underline{V})^T]\}
\end{aligned} \tag{16}$$

After some manipulation, the weighting matrix $\underline{W}$ which minimizes Eq. (16) can be shown to be

$$\begin{aligned}
\underline{W} = \underline{C} \cdot E\{&\underline{b}\,\underline{R}^{-1}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T + \underline{b}\,\underline{R}^{-1}\,\underline{V}^T\} \cdot \\
[E\{&\underline{S}\,\underline{C}\,\underline{b}\,\underline{R}^{-1}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T + \underline{S}\,\underline{C}\,\underline{b}\,\underline{R}^{-1}\,\underline{V}^T + \\
&\underline{V}\,\underline{R}^{-1}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T + \underline{V}\,\underline{R}^{-1}\,\underline{V}^T\}]^{-1}
\end{aligned} \tag{17}$$

The inclusion of $\underline{R}^{-1}$ in Eqs. (14), (16), and (17) allows the perceptual effects of errors to be accounted for; that is, a transmission error in one region of an image may be much more (or less) objectionable than a transmission error in another region. Using $\underline{R}^{-1}$, these effects can be modeled.

To simplify Eq. (17) further, a particular choice for $\underline{R}^{-1}$ must be made. A common choice, which ignores perceptual effects, is to let $\underline{R}^{-1} = \underline{I}$, so that Eq. (17) becomes

$$\begin{aligned}
\underline{W}_{opt} = \underline{C}\,E\,\{&\underline{b}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T + \underline{b}\,\underline{V}^T\} \cdot \\
[E\{&\underline{S}\,\underline{C}\,\underline{b}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T + \underline{S}\,\underline{C}\,\underline{b}\,\underline{V}^T + \underline{V}\,\underline{b}^T\,\underline{C}^T\,\underline{S}^T \\
&+ \underline{V}\,\underline{V}^T\}]^{-1}
\end{aligned} \tag{18}$$

The weighting matrix $\underline{W}$ in Eq. (18) can be written in many different forms. One particularly useful form can be obtained by defining

$$\underline{Z} = \tilde{\underline{b}}_T + \underline{V} = \underline{S}\ \underline{C}\ \underline{b} + \underline{V} \tag{19}$$

and by letting

$$\Sigma_{bz} = E\{\underline{b}\ \underline{Z}^T\} \tag{20}$$

and

$$\Sigma_z = E\{\underline{Z}\ \underline{Z}^T\} \tag{21}$$

so that Eq. (18) becomes

$$\underline{W}_{opt} = \underline{C}\ \Sigma_{bz}\ \Sigma_z^{-1} \tag{22}$$

The minimum mean squared error for this weighting matrix follows directly from Eq. (16) as

$$\varepsilon^2_{opt} = \mathrm{tr}\ [\Sigma_b - \Sigma_{bz}\ \Sigma_z^{-1}\ \Sigma_{bz}^T] \tag{23}$$

where $\Sigma_b = E\{\underline{b}\ \underline{b}^T\}$.

It is of interest to compare the performance of W_{opt} to two other suboptimal weighting matrices. First, consider the case where W is chosen under the assumption that there is no noise at all. Then,

$$\underline{W}_{s01} = [\underline{C}\ \Sigma_b\ \underline{C}^T\ \underline{S}^T][\underline{S}\ \underline{C}\ \Sigma_b\ \underline{C}^T\ \underline{S}^T]^{-1} \tag{24}$$

but if there is actually noise present that is correlated with the vector $\underline{S}\ \underline{C}\ \underline{b}$, then $\underline{W}_{s01}$ achieves the mean squared error

$$\varepsilon^2_{s01} = \mathrm{tr}\ [\Sigma_b + (\underline{C}^{-1}\ \underline{W}_{s01} - \Sigma_{bz}\ \Sigma_z^{-1})\Sigma_z\ \cdot$$

$$(\underline{C}^{-1}\ \underline{W}_{s01} - \Sigma_{bz}\ \Sigma_z^{-1})^T - \Sigma_{bz}\ \Sigma_z^{-1}\ \Sigma_{bz}^T] \tag{25}$$

where it should be noted that $\Sigma_{bz}\ \Sigma_z^{-1} = \underline{C}^{-1}\ \underline{W}_{opt}$

Another case of interest is where $\underline{W}$ is calculated under the assumption that the additive noise is uncorrelated with $\underline{S}\ \underline{C}\ \underline{b}$ so

$$\underline{W}_{s02} = [\underline{C}\ \Sigma_b\ \underline{C}^T\ \underline{S}^T][\underline{S}\ \underline{C}\ \Sigma_b\ \underline{C}^T\ \underline{S}^T + \Sigma_v]^{-1} \tag{26}$$

which produces the mean squared error

$$\varepsilon^2_{s02} = \mathrm{tr}\ [\Sigma_b + (\underline{C}^{-1}\ \underline{W}_{s02} - \Sigma_{bz}\ \Sigma_z^{-1})\ \Sigma_z\ \cdot$$

$$(\underline{C}^{-1}\ \underline{W}_{s02} - \Sigma_{bz}\ \Sigma_z^{-1})^T - \Sigma_{bz}\ \Sigma_z^{-1}\ \Sigma_{bz}^T] \tag{27}$$

Since the trace operation is distributive, it is clear from Eqs. (25) and (27) that

$$\varepsilon^2_{opt} \leq \varepsilon^2_{s01} \text{ and } \varepsilon^2_{opt} \leq \varepsilon^2_{s02}.$$

IV. EXAMPLES

It is instructive to consider a few examples in order to establish the magnitude of the performance improvement available. In each of the following examples an 8-vector is operated on by the discrete cosine transform (DCT) to yield a coefficient vector that has an approximately diagonal covariance matrix with elements given in Table III. The 8-vector is to be encoded at 1 bit/element, and the bit allocation that minimizes the mean squared error is also included in Table III.

Example 1:*

Consider the case where $\underline{V}$ represents quantization noise alone. In this case, there is no channel noise, and the soft decision demodulation scheme would not be employed. However, it is of interest to ask if the filter $\underline{W}$ might aid in the reduction of quantization noise power. The mean squared quantization noise power for each of the coefficients is given in Table III; hence

$$\Sigma_V = \mathrm{diag}\,[.05875 \quad .1182 \quad .1257 \quad .0603] \tag{28}$$

Further,

$$\underline{C}\Sigma_b\underline{C}^T = \mathrm{diag}\,[6.186 \quad 1.006 \quad .346 \quad .166 \quad .105 \quad .076 \quad .064 \quad .055] \tag{29}$$

and

$$\underline{S}\ \underline{C}\ \Sigma_b\ \underline{C}^T\ \underline{S}^T = \mathrm{diag}\,[6.186 \quad 1.006 \quad .346 \quad .166] \tag{30}$$

Now from Bucklew and Gallagher [8], if s is the input to a minimum mean squared error quantizer and n_q represents quantization noise, then

$$E\{sn_q\} \leq -\,0.5\,E\{n_q^2\} \tag{31}$$

and hence $\Sigma_{bV} \neq [0]$ but is given by

$$C\Sigma_{bV} = \begin{bmatrix} \rho_{bv1} & 0 & 0 & \cdots & 0 \\ 0 & \rho_{bv2} & 0 & & \vdots \\ 0 & 0 & \ddots & & 0 \\ \vdots & & & \ddots & \\ 0 & \cdots & \cdots & \cdots & \rho_{bvm} \\ 0 & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & & & & 0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} n^2-m \end{matrix} \tag{32}$$

Assuming

$$\rho_{bvi} = -\,0.5\,E\{v_i^2\} \tag{33}$$

Σ_{bV} can be calculated. Noting that for a unitary transform, $\mathrm{tr}[\underline{C}\ \underline{A}\ \underline{C}^{-1}] = \mathrm{tr}[\underline{A}]$, the minimum mean squared error can be found from Eq. (23) to be

$$\varepsilon^2_{opt} = 0.6425$$

The mean squared error for this case when $\underline{W}_{s01}$ in Eq. (24) is used is

$$\varepsilon^2_{s01} = 0.66295$$

The reduction obtained by modeling the noise is thus 3% of ε^2_{s01}. Considering quantization noise alone (ignoring truncation error), the reduction in quantization noise power is about 5.5%.

Example 2:*

For this example, both quantization and channel noise are included in the analysis. The channel noise contribution to Σ_V is additive to each of the diagonal elements with the magnitude of the noise power dependent on the A-factors and the channel probability of error P_e. For the purposes of this example, let $P_e = 10^{-2}$ and write

$$\sum_V = \sum_{V_1} + \sum_{V_2} \qquad (34)$$

where $\sum_{V_1}$ is due to quantization noise, given by Eq. (28), and $\sum_{V_2}$ is due to channel noise alone. $\sum_{V_2}$ is diagonal, with each of the elements given by

$$\xi_{ii} = P_e \, \sigma_i^2 \sum_{j=1}^{N} A_j$$

where σ_i^2 is the energy in the coefficient and N = number of bits. Thus,

$$\sum_{V_2} = \text{diag}\ [.3929 \quad .0468 \quad .0088 \quad .0042]$$

The minimum mean squared error using $\underline{W}_{opt}$ in Eq. (22) is thus

$$\varepsilon_{opt}^2 = 1.0569$$

The weighting matrix $\underline{W}_{s01}$ yields an error power of

$$\varepsilon_{s01}^2 = 1.1157$$

Therefore, the optimal weighting matrix reduces the mean squared error with respect to ε_{s01}^2 by about 5.3%.

V. CONCLUSIONS

A method is presented to reduce the effects of channel errors on an image transform coding system without using error correcting codes. In this method, the received transform coefficient is smoothed only if a soft decision demodulation scheme indicates that the coefficient is unreliable. An optimal memoryless reconstruction matrix was derived, and its' performance was evaluated in a simple example. While the soft decision demodulation approach seems promising, smoothers with memory will probably be necessary to make the overall system attractive.

REFERENCES

[1] P. A. Wintz, "Transform picture coding," Proc. IEEE, vol. 60, pp. 809-820, July 1972.

[2] J. M. Wozencraft and I. M. Jacobs, Principles of Communication Engineering, New York: John Wiley, 1965.

[3] C. W. Sundberg, "Soft decision demodulation for PCM encoded signals," IEEE Trans. Commun., vol. COM-26, pp. 854-859, June 1978.

[4] N. Rydbeck and C.-E. W. Sundberg, "Analysis of Digital Errors in Nonlinear PCM Systems," IEEE Trans. Commun., vol. COM-24, pp. 59-65, Jan. 1976.

[5] J. Max, "Quantizing for minimum distortion," IRE Trans. Inform. Theory, vol. IT-6, pp. 7-12, March 1960.

[6] W. K. Pratt, "Transform image coding spectrum extrapolation," Proc., Hawaii Int. Conf. on Syst. Sciences, pp. 7-9, Jan 1974.

[7] E. L. Hall, Computer Image Processing and Recognition, New York: Academic Press, 1979.

[8] J. A. Bucklew and N. C. Gallagher, "A note on optimal quantization," IEEE Trans. Inform. Theory, vol. IT-25, pp. 365-366, May 1979.

---

* For these examples, the dc coefficient is treated as if it were Gaussianly-distributed.

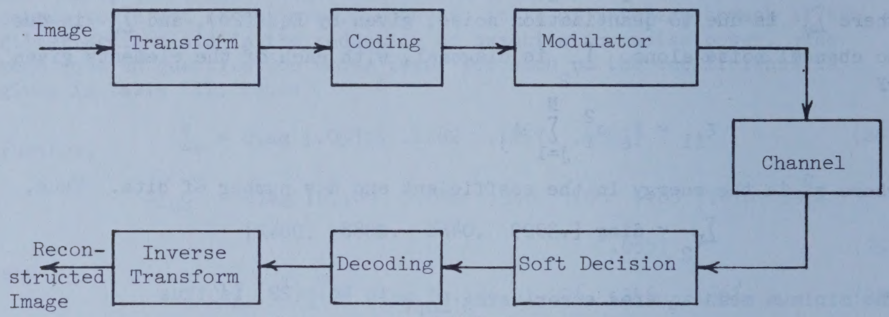


Image
Transform
Coding
Modulator
Channel
Soft Decision
Decoding
Inverse Transform
Recon-structed Image

Fig. 1 Overall System

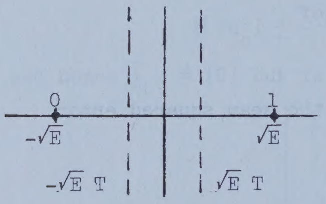

0
1
$-\sqrt{E}$
$\sqrt{E}$
$-\sqrt{E}$ T
$\sqrt{E}$ T

Fig. 2 Soft Decision Concept

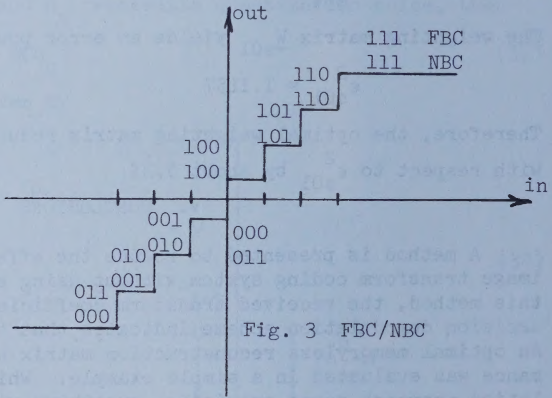

out
in
111 FBC
111 NBC
110
110
101
101
100
100
000
011
001
010
010
001
011
000

Fig. 3 FBC/NBC

Table I

8-Bit A-Factors for the FBC, NBC, and MDC

Bit in Error	Error Sequence	FBC A-Factors	NBC A-Factors	MDC A-Factors
1	10000000	3.98125	8.99989	3.98125
2	01000000	2.24998	2.24998	0.60648
3	00100000	0.56250	0.56250	0.57118
4	00010000	0.14062	0.14062	0.53751
5	00001000	0.03506	0.13516	0.48732
6	00000100	0.00877	0.00877	0.40813
7	00000010	0.00220	0.00220	0.28575
8	00000001	0.00055	0.00055	0.11975
Sum of A-Factors=		6.98103	11.99967	6.99737

Table II
Normalized Soft Decision Thresholds,
N = 8, M = 3,
Gaussian pdf

	Number of Bits (N) 1	2	3	←→ 8
T_1	0.35 ←			→ 0.35
T_2	X	0.29 ←		→ 0.29
T_3	X	X	0.15 ←	→ 0.15

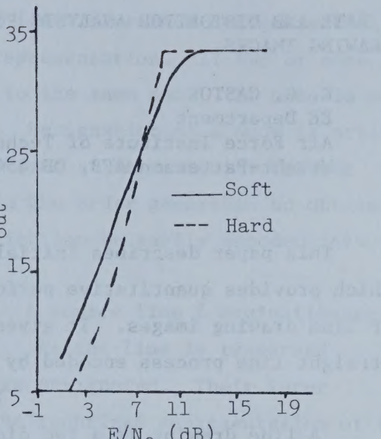


Fig. 4 Soft vs. Hard Decision

Table III
Energy and Bit Allocation for DCT Coefficients

Coefficient	Variance	No. of Bits	Quantization Noise Power
1	6.186	4	0.05875
2	1.006	2	0.118205
3	0.346	1	0.12574
4	0.166	1	0.06032
5	0.105	0	0.105
6	0.076	0	0.076
7	0.064	0	0.064
8	0.055	0	0.055

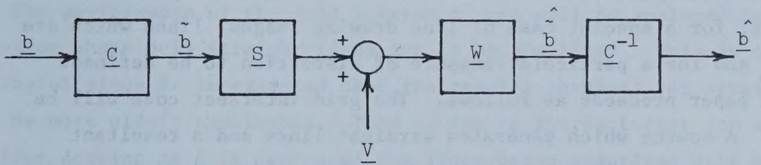

$\underline{b}$
$\underline{C}$
$\tilde{\underline{b}}$
$\underline{S}$
+
+
$\underline{V}$
$\underline{W}$
$\hat{\tilde{\underline{b}}}$
$\underline{C}^{-1}$
$\hat{\underline{b}}$

Fig. 5 Data Vector Processing

A RATE AND DISTORTION ANALYSIS FOR GRID INTERSECT QUANTIZATION OF LINE DRAWING IMAGES

K. G. CASTOR
EE Department
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433

D. L. NEUHOFF
Elec. and Comp. Eng Dept
University of Michigan
Ann Arbor, MI 48109

ABSTRACT

This paper describes initial effort in the development of a theory which provides quantitative performance measures for grid based quantization of line drawing images. It gives analysis and asymptotic results for a straight line process encoded by the grid intersect code.

INTRODUCTION

A line drawing is a two dimensional image wherein information is conveyed by the shape and location of thin lines on a contrasting background. Examples of such drawings include charts, graphs, maps and alphanumeric text. A number of digital quantization techniques have been devised specifically for processing members of this class of image data. Referred to collectively as grid based quantization methods, these schemes involve the overlaying of a square grid on the line drawing image and then selecting and interconnecting particular nodes of the grid as the quantized representation. H. Freeman is responsible for much of the developmental effort in grid based quantization of line drawing images [1] - [3].

The subject of grid based quantization of line drawing images lacks a theory which provides quantitative measures of performance. This paper presents the results of initial efforts to develop such a theory. The performance analysis will be presented for a single code, the grid intersect code, for a special case of line drawing images, lines which are straight and for a particular measure of distortion to be defined.

The paper proceeds as follows. The grid intersect code will be defined. A source which generates straight lines and a resultant segmented line process will then be described. After defining performance measures, asymptotic results will be obtained for the case where the straight lines are allowed to attain infinite length.

THE GRID INTERSECT CODE

Assume that quantization is being accomplished with respect to a grid system which is a lattice of squares each having sides of dimension Δ. The grid intersect code [2] is one of the simplest quantization schemes. The rule for selection and interconnection of grid nodes to obtain a quantized representation of a line is as follows. Traverse the source line from its starting point. Each time a grid line is crossed designate the closest node as a point of the quantized representation. If two or more successive grid line crossings correspond to the same node, the node is designated only once for the first crossing. Designation of a node is arbitrary in the case where the source line crosses at the midpoint between two nodes. The nodes are interconnected in the order generated to obtain a quantized version of the source line, which can be easily encoded into a binary representation.

The process is illustrated in Fig. 1. A source line ℓ orginating at point p_0' is shown with a superimposed grid. As the line is traversed from p_0' grid nodes g_0, g_1, g_2, g_3 and g_4 are designated. Their interconnection, shown by the dashed line, is the quantized representation of the portion of ℓ shown.

Once any node g_k is specified, the next node of the quantization g_{k+1} must be one of the eight nodes nearest to g_k. Similarly, the next portion of the quantization is a ray from g_k to g_{k+1}. The eight possible rays between g_k and g_{k+1} are shown in Fig. 2 labeled $\tilde{c}_1$ to $\tilde{c}_8$. If $\mathcal{C} = \{\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_8\}$, then $\mathcal{C}$ may be thought of as a codebook. The quantization $\hat{\ell}$ of a line ℓ is merely a concatenation of code letter elements of $\mathcal{C}$ properly located in the grid. To illustrate, the quantization of ℓ shown in Fig. 1 may be transmitted or stored by specifying the location of g_0 and the sequence of code letters $\tilde{c}_2 \circ \tilde{c}_1 \circ \tilde{c}_8 \circ \tilde{c}_1$. Since there are 8 possible codewords in the codebook, each codeletter can be specified using three bits to obtain a binary representation of the quantization.

A STRAIGHT LINE SOURCE

The performance of the grid intersect code will be analyzed for the situation where only straight lines are to be quantized. This is considered meaningful since it is expected that the results obtained for straight lines will be more widely applicable. This is due to the fact that for almost any line drawing as Δ is decreased the lines being quantized will become relatively straight over distances of many multiples of Δ. Thus for small Δ, which will be the case for high quality quantization, the straight line problem is highly significant.

The probabilistic model we will adopt for a straight line source will now be described. Let L_s be the set of all directed straight lines with starting points in the plane. Each line $\ell \epsilon L_s$ can be characterized by its starting point $p_0' = (x_0', y_0')$ and its angles of emanation θ relative to some fixed direction. (Assume a pair of coordinate axes where the x-axis is a horizontal grid line and the y-axis a vertical grid line). We assume that p_0' has a distribution that is uniform over each grid square and that θ is independent of p_0' and uniformly distributed over a 2π radian interval, say $[-\frac{\pi}{2}, \frac{3\pi}{2})$. While this model is certainly simplistic, the assumption about the distribution of p_0' is good when Δ is small and p_0' actually has some nonatomic distribution. The assumption that θ is uniformly distributed and independent is valid for drawings that have roughly equal proportions of line segments at all angles, for situations where the orientation of the drawing is itself random or for the case where the orientation of the quantizing grid is random. Furthermore, the model is tractable.

A SEGMENTED LINE PROCESS

We can now take advantage of certain geometrical properties of straight lines in a grid system to allow straightforward analysis of the performance of the grid intersect code on the straight line model. Observe that for straight lines with $\theta \varepsilon [-\frac{\pi}{4}, \frac{\pi}{4}) \cup [\frac{3\pi}{4}, \frac{5\pi}{4})$, the grid intersect encoding algorithm can be redefined as follows. Traverse the source line from its starting point. Each time a vertical grid line is crossed, designate the closest node as a point of the quantization. That is, only the vertical grid line crossing points are necessary to determine the quantized representation of the line. Similarly, for other values of θ only the horizontal grid line crossing points are necessary to generate a quantized representation. This may be easily verified through experimentation with the grid intersect quantization procedure.

Hence, we now describe a process which characterizes the sequence of crossing points of the source line with "appropriate" grid lines. As the source line ℓ is traversed from p_0', if $\theta \varepsilon [-\frac{\pi}{4}, \frac{\pi}{4}) \cup [\frac{3\pi}{4}, \frac{5\pi}{4})$, let $p_0 = (x_0, y_0)$ be the first vertical grid line crossing point. Otherwise, let p_0 be the first horizontal grid line crossing point. Now, g_0 is a node which is at a distance of no greater than $\frac{\Delta}{2}$ from p_0. Similarly, if p_{k-1} is the kth vertical or horizontal grid line crossed, then g_{k-1} is within $\frac{\Delta}{2}$ of p_{k-1}.

Let ℓ_k correspond to the segment of ℓ which lies between the (k-1)th and kth grid lines crossed. The quantization $\hat{\ell}_k$ of ℓ_k is merely the code letter $c_k \varepsilon C$. Furthermore, c_k may be determined from knowledge of only p_{k-1} and θ. In fact, if positive and negative distance corresponds to direction on the reference coordinate axes, then knowledge of the signed distance of p_{k-1} from g_{k-1} and the angle θ is sufficient to determine p_k, g_k and c_k.

A unique form of modulus arithmetic will be used to determine the signed distance of p_k from g_k. Each point p_k is specified by its coordinates (x_k, y_k) with respect to the reference axes. Define

$$w \text{ modulo } [\pm\frac{\Delta}{2}] = \begin{cases} w \bmod \Delta & \text{if } w \bmod \Delta \leq \frac{\Delta}{2} \\ w \bmod \Delta - \Delta & \text{otherwise} \end{cases} \tag{1}$$

Then, the signed distance desired is given by

$$p^*_k = \begin{cases} y_k \bmod [\pm\frac{\Delta}{2}] & \text{if } \theta\varepsilon[-\frac{\pi}{4},\frac{\pi}{4}) \cup [\frac{3\pi}{4},\frac{5\pi}{4}) \\ x_k \bmod [\pm\frac{\Delta}{2}] & \text{otherwise} \end{cases} \tag{2}$$

For each $\ell\varepsilon L_s$ a sequence of points $p^*_k(\ell) = p^*_k$, $k = 0,1,2,\ldots$ may be generated. Furthermore, it may be shown that this sequence, $\{p^*_k\}^{\infty}_{k=0}$, is a stationary random process and conditioned on a specific value of θ, such that tan θ is irrational, $\{p^*_k\}^{\infty}_{k=0}$ is stationary and ergodic.

PERFORMANCE MEASURES

Analagous to more traditional coding theory, both distortion and rate measures will be defined for performance analysis of the grid intersect quantization of line drawing images. The distortion measure chosen is a length normalized measure of area between the line and its quantization. Similarly, the rate measure is the length normalized number of bits in the binary quantized representation. Each of these quantities will be defined below.

The length normalized area error between a line $\ell\varepsilon L_s$ and its quantization $\hat{\ell}$ will be obtained by summing incremental portions of normalized area. Each area increment corresponds to a segment ℓ_k of ℓ's segmentation as described in the previous section. For a segment ℓ_k with representation c_k, let $\rho(p^*_{k-1},\theta)$ be the area between ℓ_k and c_k bounded by the vertical grid lines which contain g_{k-1} and g_k. The shaded portion of Fig. 3 illustrates $\rho(p^*_0,\theta)$.

With the length of the segment ℓ_k given by

$$|\ell_k| = \begin{cases} \dfrac{\Delta}{\cos\theta} & \theta\varepsilon[-\frac{\pi}{4},\frac{\pi}{4}) \\ \dfrac{\Delta}{\sin\theta} & \theta\varepsilon[\frac{\pi}{4},\frac{3\pi}{4}) \\ \dfrac{\Delta}{-\cos\theta} & \theta\varepsilon[\frac{3\pi}{4},\frac{5\pi}{4}) \\ \dfrac{\Delta}{-\sin\theta} & \text{otherwise} \end{cases} \tag{3}$$

define the increment of area error associated with ℓ_k as

$$A_k = \frac{\rho(p^*_{k-1},\theta)}{|\ell_k|} \tag{4}$$

Next, the asymptotic straight line distortion associated with representing ℓ by $\hat{\ell}$ is defined by

$$D_s(\ell,\hat{\ell}) = \lim_{K\to\infty} \sup \frac{1}{K} \sum_{k=1}^{K} A_k \tag{5}$$

Finally, the average distortion associated with quantization of straight lines using the grid intersect method is

$$\mathcal{D}_s = E_{p_0^*,\theta}\{D_s(\ell,\hat{\ell})\} \tag{6}$$

A number of relevant observations may be made regarding equations (5) and (6). Since the $\{p_k^*\}_{k=0}^{\infty}$ sequence is stationary, the sequence $\{A_k\}_{k=0}^{\infty}$ is stationary and it follows that the limit in (5) exists almost surely. In fact, conditioned on a specific value of θ such that $\tan\theta$ is irrational, the stationary, ergodic property of the $\{p_k^*\}_{k=0}^{\infty}$ sequence implies that the limit in (5) depends only on θ and (5) reduces to

$$D_s(\ell,\hat{\ell}) = E_{p_0^*}\{A_1|\theta\} \triangleq \psi(\theta) \text{ a.s.} \tag{7}$$

and therefore, (6) becomes

$$\mathcal{D}_s = E_{\theta}\{\psi(\theta)\} \tag{8}$$

The significance of the conditional ergodicity of $\{p_k^*\}_{k=0}^{\infty}$ is that regardless of the location of the initial point of the line $\ell \varepsilon L_s$, $D_s(\ell,\hat{\ell})$ for a given θ with irrational tangent will with probability one converge to $\psi(\theta)$.

The rate measure is similarly defined. Let $b(c_k)$ be the number of bits required to represent quantization letter c_k where $c_k \varepsilon C$. The rate measure associated with quantization of a line is defined as the length normalized total number of bits in the representation given by

$$R(\ell,\hat{\ell}) = \lim_{K\to\infty} \sup \frac{1}{K} \sum_{k=1}^{K} \frac{b(c_k)}{|\ell_k|} \tag{9}$$

The average asymptotic straight line rate for the grid intersect code is given by

$$\mathcal{R}_s(\text{grid intersect}) = E_{p_0^*,\theta}\ \{R(\ell,\hat{\ell})\} \tag{10}$$

DISTORTION ANALYSIS

Equation (8) may be further simplified by recognizing certain symmetry properties of the grid intersect code. For example, if the quantization of ℓ is $\hat{\ell} = Q(\ell)$ and if $r_\phi(\ell)$ is a transformation which rotates ℓ about g_0 by ϕ degrees then

$$Q[r_{n\frac{\pi}{2}}(\ell)] = r_{n\frac{\pi}{2}}[Q(\ell)] \qquad n = \pm1,\pm2,\pm3 \tag{11}$$

Furthermore, if $T_m(\ell)$ is the transformation which carries ℓ into its mirror image through the grid parallel to the x-axis which contains g_0 then

$$Q[T_m(\ell)] = T_m[Q(\ell)]. \tag{12}$$

Using (3), (11) and (12), (8) becomes

$$\mathcal{D}_s = \frac{4}{\pi}\int_0^{\frac{\pi}{4}}\psi(\theta)d\theta \tag{13}$$

where

$$\psi(\theta) = \frac{1}{\Delta^2}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \rho(s,\theta)\cos\theta d\theta \quad (\text{letting } s = p_0^*) \tag{14}$$

Computation of (14) involves consideration of θ in two ranges, $[0,\tan^{-1}\frac{1}{2})$ and $[\tan^{-1}\frac{1}{2}, \frac{\pi}{4})$, and s in four ranges for each θ range. The process will be illustrated for one range of θ.

For $\theta\varepsilon[0,\tan^{-1}\frac{1}{2})$ and referring to Fig. 4, $\rho(s,\theta)$ takes on the following values:

$$\rho(s,\theta) = \begin{cases} -\frac{1}{2}\Delta^2\tan\theta-\Delta s & s\varepsilon[-\frac{\Delta}{2}, -\Delta\tan\theta] \\ \frac{s^2}{\tan\theta} + s\Delta + \frac{1}{2}\Delta^2\tan\theta & s\varepsilon[-\Delta\tan\theta,0] \\ s\Delta + \frac{1}{2}\Delta^2\tan\theta & s\varepsilon[0,\frac{\Delta}{2} -\Delta\tan\theta] \\ \frac{s^2}{1-\tan\theta} - s\Delta + \frac{1}{2}\Delta^2(1-\tan\theta) & s\varepsilon[\frac{\Delta}{2} - \Delta\tan\theta,\frac{\Delta}{2}] \end{cases} \tag{15}$$

Performing the integration (14)

$$\psi(\theta) = \frac{1}{6}\Delta\cos\theta - \frac{1}{3}\Delta\sin\theta + \frac{\Delta\cos\theta}{12(1-\tan\theta)} \qquad \theta\varepsilon[0,\tan^{-1}\frac{1}{2}) \tag{16}$$

A similar computation yields

$$\psi(\theta) = -\frac{1}{6}\Delta\cos\theta + \frac{1}{3}\Delta\sin\theta + \frac{\Delta\cos^2\theta}{12\sin\theta} \qquad \theta\varepsilon[\tan^{-1}\frac{1}{2},0) \tag{17}$$

Finally, integrating (16) and (17) over their respective ranges yields

$$\mathcal{D}_s = .17871\Delta \, \frac{\text{square units}}{\text{unit length}} \tag{18}$$

RATE ANALYSIS

For the grid intersect code $b(c_k) = 3$. Thus, using (3), (9), (10), (11) and (12) the average asymptotic straight line rate is given by

$$R_s(\text{grid intersect}) = \frac{1}{2\pi}\int_{-\frac{\pi}{2}}^{\frac{3\pi}{2}} [\frac{1}{\Delta}\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \frac{3}{|\ell_k|}\, ds]d\theta$$

$$= \frac{6\sqrt{2}}{\pi\Delta} = \frac{2.70095}{\Delta}\, \frac{\text{bits}}{\text{unit length}} \tag{19}$$

CONCLUSION

Having defined rate and distortion measures, we have illustrated computation of these performance figures for a straight line process where quantization is accomplished using the grid intersect code. Our results (18) and (19) yield $\mathcal{D}_s R_s = .46405$.

REFERENCES

1. H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", IRE Trans on Elec. Comp., Vol EC-10, pp. 260-268, June 1961.

2. H. Freeman, "Computer Processing of Line Drawing Images", Computing Surveys, vol. 6, no. 1, pp. 57-97, Mar. 1974.

3. H. Freeman and J. Glass, "On the Quantization of Line Drawing Data", IEEE Trans. on Sys., Sci., and Cyb., vol. SSC-5, pp. 70-79, Jan. 1969.

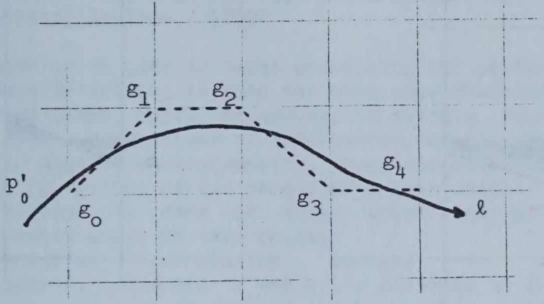

g_1
g_2
p'_0
g_4
g_0
g_3
ℓ

Fig. 1 Quantization Example

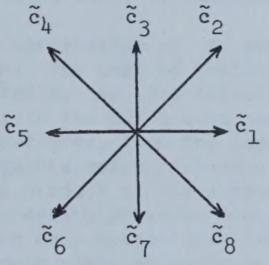

$\tilde{c}_4$
$\tilde{c}_3$
$\tilde{c}_2$
$\tilde{c}_5$
$\tilde{c}_1$
$\tilde{c}_6$
$\tilde{c}_7$
$\tilde{c}_8$

Fig. 2 Codebook

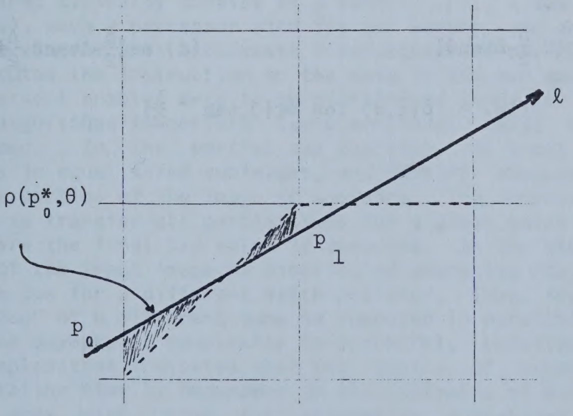
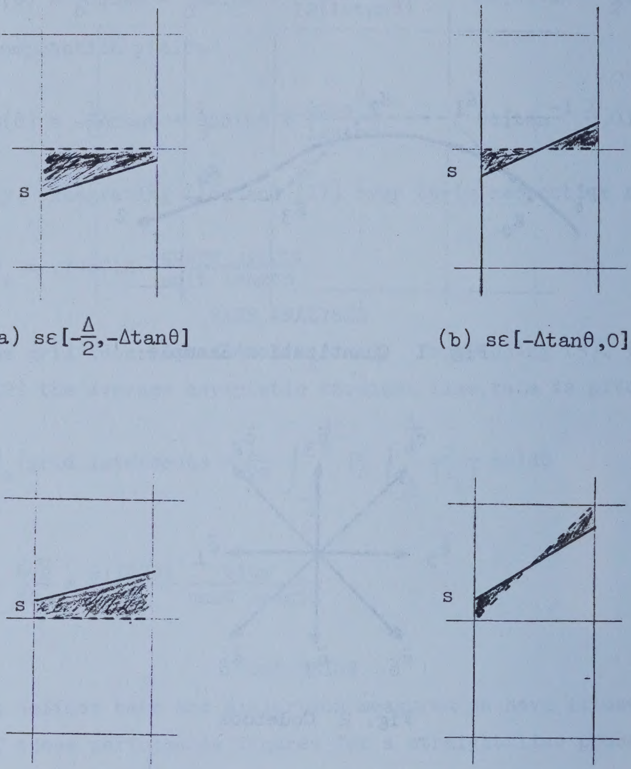

ℓ
$\rho(p^*_0,\theta)$
p_1
p_0

Fig. 3 Illustration of Area Error Increment



(a) $s\varepsilon[-\frac{\Delta}{2},-\Delta\tan\theta]$

(b) $s\varepsilon[-\Delta\tan\theta,0]$

(c) $s\varepsilon[0,\frac{\Delta}{2}-\Delta\tan\theta]$

(d) $s\varepsilon[\frac{\Delta}{2}-\Delta\tan\theta,\frac{\Delta}{2}]$

Fig. 4 $\rho(s,\theta)$ for $\theta\varepsilon[0,\tan^{-1}\frac{1}{2}]$

SIMD ALGORITHMS TO PERFORM TWO-DIMENSIONAL IMAGE CORRELATION

ARTHUR E. FEATHER & LEAH J. SIEGEL
School of Electrical Engineering, Purdue University
West Lafayette, Indiana 47907

Image correlation is used in image processing to perform a variety of tasks: in edge detection, to find the step edge discontinuity between two areas, or to find lines, spots, or curves; to match a representation of a known object to a given frame of information, or to match two pictures to one another. In digital photogrammetry, image correlation is used to locate the corresponding points of two images of a stereomodel. In this application, the correlation is done on areas which typically are at least 4096x4096 with match areas of size 64x64.

A representative correlation measure is the covariance $S(XY) = \Sigma xy - \Sigma x \Sigma y/n$. Symbols x and y are elements of X and Y, where X is the matching image and Y is an area of the input image which is the same size as X; n is the number of elements in the match area. To correlate a match area with a larger image, the match area is positioned at each possible location on the image, and the value of $S(XY)$ is computed for each such position. The area Y for which $S(XY)$ is maximum corresponds to the best match.

Implementation of image correlation on a serial processor requires a great deal of computation. The cost of computing the $S(XY)$ values over an input image will be dominated by the Σxy calculations. If r and c are the number of rows and columns in the match area, and R and C are the number of rows and columns in the input area, then the asymptotic complexity of computing Σxy for all possible match positions will be $O(rcRC)$, that is it will be proportional to the product of the areas of the match area and the input image. Parallel processing provides one means of reducing the computation time, since there are many operations involved in the correlation procedure that can be done simultaneously in different processors. SIMD (single instruction stream-multiple data stream) machines represent one type of parallel processing which is especially suitable to matrix operations.

SIMD machines typically consist of a control unit, a set of N processing elements (PEs), each a processor with its own memory, and an interconnection network. The control unit broadcasts instructions to all PEs, and each active PE executes the instruction on the data in its own memory. The interconnection network enables data to be transferred among PEs.

Two SIMD algorithms to perform two-dimensional image correlation have been developed. In the partial sum approach, the input image is divided among the PEs in equal sized subimages, and each PE computes partial sums involving the portion of the image it contains. The interconnection network is then used to transfer all partial sums for a given match position to the same PE, where the final Σxy value is computed. In the group sum approach, each column of the input image is distributed among the PEs, and each PE accumulates the sum for a different match position. Thus, for a system having N PEs, a "group" of N different sums is computed in parallel. For both algorithms, the asymptotic complexity is $O(rcRC/N)$. Detailed analysis of the algorithm complexities indicates that the choice of algorithm to reduce overall operation time is dependent on the variables of machine size, image size, match area size, time for arithmetic operations, and time for transfers.

This work was supported by the DMA, monitored by the USAF RADC Info. Sci. Div., under Contract No. F30602-78-C-0025 through Univ. Mich.

SIMULTANEOUS LINEAR ALGEBRAIC EQUATION FORMULATIONS OF 2D FILTERS

D. E. OLSON, W. E. ALEXANDER, E. E. SHERROD
Elec. Eng. Dept., NC A&T SU, Greensboro, NC

ABSTRACT

It is shown that 2D digital filter realizations are equivalent to the solution of tensor equations, and they are also equivalent to the solution of matrix equations. Both recursible and non-recursible filters are included in these formulations.

SUMMARY

A 2D digital filter, which possesses a rational transfer function, may be represented by its bivariate difference equation written in tensor form as:

$$b_{ij}\, g_{p+i,q+j} = a_{ij}\, f_{p+i,q+j} \tag{1}$$

where $1 \leq p \leq N$, $1 \leq q \leq M$, $-m \leq i \leq m$, $-m \leq j \leq m$; and the double appearance of an indice on a given side of the equality implying the usual summation over the appropriate range of that indice. A more formal expression of (1) is:

$$B^{kl}_{pq}\, g_{kl} = A^{kl}_{pq}\, f_{kl} \tag{2}$$

where $1 \leq k \leq N$, $1 \leq l \leq M$, and the non-zero components of the coefficient tensors given by $A^{kl}_{pq} = a_{k-p,l-q}$; and $B^{kl}_{pq} = b_{k-p,l-q}$; for $-m \leq k-p \leq m$, $-m \leq l-q \leq m$.

The 2D filtering operation requires that one determine all the g_{pq}, given all a_{ij}, b_{ij}, and f_{pq}. A solution will exist and be unique if there exists an inverse of the tensor B^{kl}_{pq}, say C^{pq}_{uv}; with $1 \leq u \leq N$, $1 \leq v \leq M$. For such a case, the filtered solution would then be given by:

$$g_{uv} = C^{pq}_{uv}\, A^{kl}_{pq}\, f_{kl} \tag{3}$$

Tensor equation (2) can also be interpreted as a matrix equation with the A^{kl}_{pq}, B^{kl}_{pq} taken as NMxNM dimensional coefficient matrices with row index "pq", column index "kl"; and g_{kl}, f_{kl} taken as column matrices.

For the case when N=M, and a_{00}, $b_{00} \neq 0$; then equation (2) is also expressible as a matrix equation involving only NxN matrices given by:

$$LGR + \sum_{k=-m,k\neq 0}^{m} S_k G\, T_k = c\ PFQ + c \sum_{k=-m,k\neq 0}^{m} S_k F\, U_k \tag{4}$$

where $c = a_{00}/b_{00}$, the matrices $G = [g_{pq}]$, $F = [f_{pq}]$; and the non-zero components of the coefficient matrices L, R, P, Q, S_k, T_k and U_k given by:

(i) For p, q such that $-m \leq q-p \leq m$:

$L_{pq} = b_{q-p,0}/b_{00}$; $R_{pq} = b_{0,q-p}/b_{00}$; $P_{pq} = a_{q-p,0}/a_{00}$; $Q_{pq} = a_{0,q-p}/a_{00}$.

(ii) For p, q such that $q-p = \pm k$:

$T_{kpq} = b_{k,p-q}/b_{00} - b_{k,0}b_{0,p-q}/b^2_{00}$; $U_{kpq} = a_{k,p-q}/a_{00} - a_{k,0}a_{0,p-q}/a^2_{00}$.

(iii) And finally, for p, q such that $q-p=k$: $S_{kpq} = 1$.

Non-recursible filters generally require solutions of the form given by (3). For recursible filters (4) simplifies allowing solution by compact schemes.

FILTER DESIGN FOR ARRAY SIGNALS

M. SIMAAN
Department of Electrical Engineering
University of Pittsburgh
Pittsburgh, PA 15261

ABSTRACT

This paper deals with the design of filters for processing data obtained from arrays of sensors used for signal measurements in a variety of applications. The objective of the filters is to eliminate undesired signals intercepting the array while at the same time preserving the desired signal that the array is attempting to measure. A constrained minimization formulation for determining the filters' coefficients is used, and the computational requirements are kept to a minimum. When compared with the widely used delay-and-sum method, this technique is shown to have a superior frequency reject response which becomes more apparent if the array has a relatively small number of sensors. An example is worked out to illustrate the effectiveness of the design.

INTRODUCTION

Linear arrays of sensors are used for measurements of signals in a variety of applications such as radar, seismic, nondestructive testing and biomedical. One of the important features of array data is that it can be phased out to preserve a desired signal and eliminate undesired interfering signals. The desired signal is typically a waveform that the array is attempting to measure and the undesired signals are other waveforms intercepting the array at different velocities and from different angles.

Consider an array of N sensors (Figure 1) sensitive to the horizontal component of signals travelling along the array. Let D be the spacing between two consecutive sensors. Let V_d be the velocity of the desired signal whose incident plane wave makes an angle Θ_d with the axis of the array and let V_u be the velocity of an undesired signal whose incident plane wave makes an angle Θ_u with the axis of the array. The output of every sensor is the superposition of the horizontal components of these two signals. A well known method for attenuation of the undesired signal is known as the "delay-and-sum" method and it essentially involves shifting the data so that signals travelling in one direction are aligned and then averaging over all channels. The frequency reject-band response of this

---

*This research was supported in part by a grant from Gulf Science and Technology Company.

method is given by the expression

$$R(f) = \frac{1}{N} \sum_{i=1}^{N} \alpha_i e^{j2\pi f(i-1)\left(\frac{D\cos\Theta_u}{V_u} - \frac{D\cos\Theta_d}{V_d} \right)} \qquad (1)$$

where α_i is the relative amplitude of the undesired signal with respect to the desired signal at the i^{th} sensor. As an example, if $N = 8$, $D = 25$ ft., $V_d = V_u = 10{,}000$ ft/sec, $\Theta_d = 0^\circ$, $\Theta_u = 180^\circ$ and $\alpha_i = 1$ for $i = 1, \ldots 8$, then the plot of $|R(f)|$ is shown in Fig. 2. This plot essentially says that the delay and sum method cannot attenuate frequencies of 0, 200, 400, etc. Hz of the undesired signal. In general, frequencies corresponding to

$$f_k = \frac{1}{\frac{D\cos\Theta_u}{V_u} - \frac{D\cos\Theta_d}{V_d}} k, \quad k = 1, 2, \ldots \qquad (2)$$

of the undesired signal cannot be attenuated, and the plot of $|R(f)|$ is periodic having $N - 1$ lobes per period. Clearly the rejection property of this method is not satisfactory especially if the frequency spectrum of the undesired signal happens to fall within the frequency bands of the main lobes and large side lobes.

Ideally, because the height of the main lobes cannot be altered, it is desired to have $|R(f)|$ as close as possible to a train of impulses [1] located at the frequencies of the main lobes given by the expression in (2). We note that this cannot be achieved by increasing N since the heights of the side lobes have a non-zero limiting value [2] given by

$$R(f_s) = \frac{1}{(s+0.5)\pi} \qquad s = 1, 2, \ldots, N-2 \qquad (3)$$

as N becomes very large. In the above expression, f_s refers to the center frequency of the s^{th} side lobe.

2. OPTIMUM FILTERS

Clearly, because of engineering and economic considerations, as well as the physical limitations imposed by expression (3), increasing the number of sensors and hence making the array longer is not a viable solution for improving the reject response of the delay and sum method. The need for other processing schemes to achieve a desired degree of rejection of undesired signals motivated the present study. Our approach involves the design of N filters, one for each sensor output, to be applied after alignment and before averaging as shown in Figure 3. The criterion for the

the derivation of the filter coefficients is to: (i) preserve exactly the desired signals and to (ii) attenuate as much as possible the undesired signals. The derivation of the filter coefficients will be done in the frequency domain. In order to satisfy criterion (i), we shall impose the following constraint [3] on the filters coefficients:

$$\sum_{i=1}^{N} F_i(f) = e^{j2\pi f\tau_o} \tag{4}$$

where $F_i(f)$ is the frequency spectrum of the i^{th} filter and τ_o is a pre-selected time constant corresponding to a desired time shift of the desired signal. After application of the filters, the frequency reject response after averaging becomes:

$$R(f) = \sum_{i=1}^{N} \alpha_i \, e^{j2\pi f(i-1)\left(\frac{D\cos\Theta_u}{V_u} - \frac{D\cos\Theta_d}{V_d}\right)} F_i(f) \tag{5}$$

In order to satisfy criterion (ii), we shall choose to minimize the function

$$E = \sum_{i=1}^{N} |R(f) - R_i(f)|^2 \tag{6}$$

The interpretation of this criterion is that after filtering, the reject response of each individual sensor output is being made as close as possible to the reject response of the average output.

The problem under consideration is therefore a constraint minimization problem with the function (6) to be minimized subject to the constraint given by equation (4). If we define the filter coefficients by

$$F_i(f) = A_i(f) + jB_i(f)\ , \ i = 1, \ldots N \tag{7}$$

and let

$$\tau_i = (i-1) \frac{D\cos\Theta_u}{V_u} - \frac{D\cos\Theta_d}{V_d} \tag{8}$$

Then equation (4) reduces to

$$R(f) = \sum_{i=1}^{N} \alpha_i \, (A_i\cos 2\pi f\tau_i + B_i\sin 2\pi f\tau_i) + j\alpha_i(B_i\cos 2\pi f\tau_i - A_i\sin 2\pi f\tau_i) \tag{9}$$

The solution of this constraint optimization problem, can be obtained using Lagrange multiplier techniques, after lengthy algebraic manipulations to be of the form:

$$\begin{aligned} A(f) &= -R(f)\mu\lambda_1(f) + S(f)\mu\lambda_2(f) \\ B(f) &= -S(f)\mu\lambda_1(f) - R(f)\mu\lambda_2(f) \end{aligned} \tag{10}$$

where $A'(f) = [A_1(f), \ldots, A_N(f)]'$; $B'(f) = [B_1(f), \ldots, B_N(f)]'$ and $\lambda_1(f)$ and $\lambda_2(f)$ are the Lagrange multipliers given by the expressions:

$$\lambda_1(f) = \frac{-[\mu' R(f)\mu] \cos 2\pi f\tau_o + [\mu' S(f)\mu] \sin 2\pi f\tau_o}{[\mu' R(f)\mu]^2 + [\mu' S(f)\mu]^2}$$

$$\lambda_2(f) = \frac{[\mu' S(f)\mu] \cos 2\pi f\tau_o + [\mu' R(f)\mu] \sin 2\pi f\tau_o}{[\mu' R(f)\mu]^2 + [\mu' S(f)\mu]^2} \qquad (11)$$

where R(f) and S(f) are NxN matrices given by:

$$R(f) = \begin{bmatrix} \frac{1-\beta}{\alpha_1^2} & & -\frac{\beta}{\alpha_i\alpha_j}\cos 2\pi f(\tau_i-\tau_j) \\ & \ddots & \\ -\frac{\beta}{\alpha_i\alpha_j}\cos 2\pi f(\tau_i-\tau_j) & & \frac{1-\beta}{\alpha_N^2} \end{bmatrix} \qquad (12)$$

and

$$S(f) = \begin{bmatrix} 0 & & \frac{\beta}{\alpha_i\alpha_j}\sin 2\pi f(\tau_i-\tau_j) \\ & \ddots & \\ \frac{\beta}{\alpha_i\alpha_j}\sin 2\pi f(\tau_i-\tau_j) & & 0 \end{bmatrix} \qquad (13)$$

and where $\mu' = [1, 1, \ldots 1]$ and $\beta = \frac{N-2}{(N-1)^2}$

The closed form expressions for the optimum filters based on (10) can be easily programmed for computer implementation. If the time domain version of the filters is desired, the inverse Fourier transforms of $F_i(f)$ can be easily handled by any FFT algorithm. It is worthwhile to mention at this stage that since $A_i(-f) = A_i(f)$ and $B_i(-f) = -B_i(f)$ which follow directly from (10), the time domain filters will have only real coefficients.

3. EXAMPLE

Consider, the 8-sensor array discussed earlier. After derivation of the filters, the reject band response of the array is now shown in Figure (4). Comparing this response to Figure (2) of the delay-and-sum method, the following improvements have been clearly achieved:

(i) A drastic reduction in the heights of all side lobes, and (ii) a drastic reduction in the width of the main lobes.

To show the effect of the filters on the data itself, we consider the signals shown in Figure (5). These signals have been generated by superimposing a desired signal, Figure (6a) on the top of an undesired signal travelling along the array. In Figure (5), the signals have been aligned along the desired direction. Figure (6b) shows the desired signal as recovered by the delay-and-sum method. Figure (6c) shows the desired signal as recovered by the optimum filters method. Clearly in this case, the optimum filters method is able to extract the desired signal almost exactly while the delay and sum method fails to do so.

REFERENCES

[1] Dudgeon, D.E., "Fundamentals of Digital Array Processing," Proc. of the IEEE, Vol 65, No. 6, June 1977, pp. 898-904.

[2] Schelkunoff, S.A., Electromagnetic Waves, New York, Van Nostrand Pub., 1943.

[3] Levin, M.J., "Maximum Likelihood Array Processing," Report #31, MIT Lincoln Laboratory, December 1964.

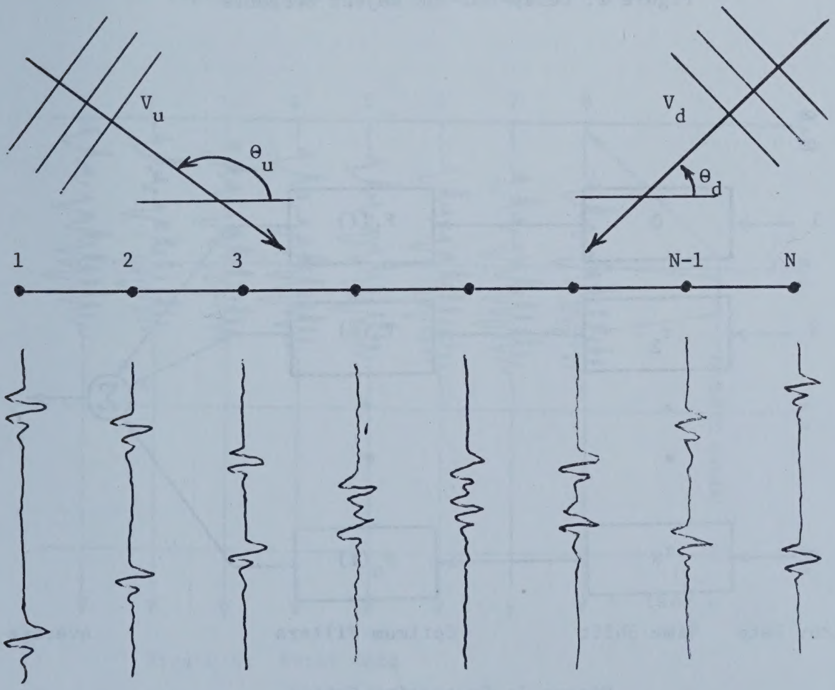


Figure 1: Array Signals

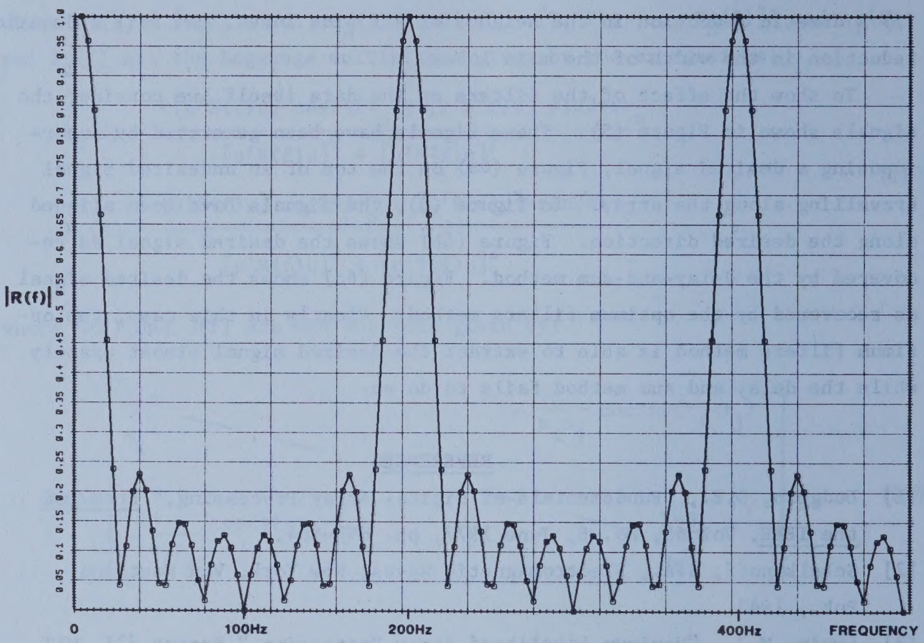


Figure 2: Delay-and-Sum Reject Response

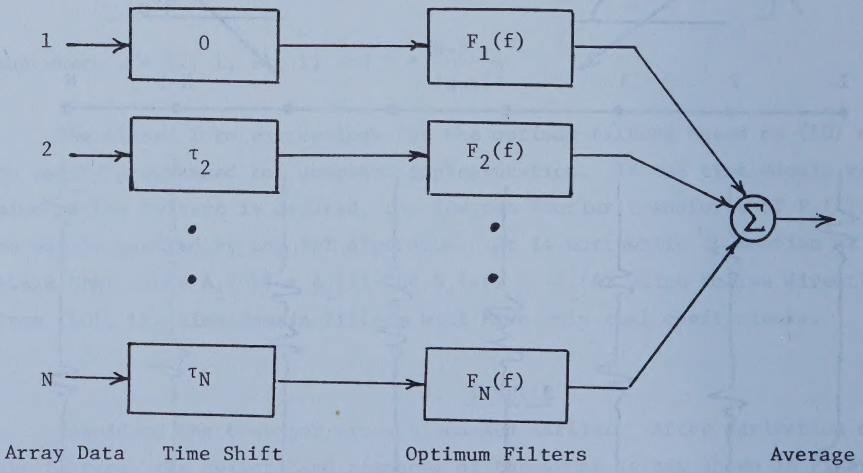

1
2
N
0
τ2
τN
F1(f)
F2(f)
FN(f)
Σ
Array Data
Time Shift
Optimum Filters
Average

Figure 3: Processing Scheme

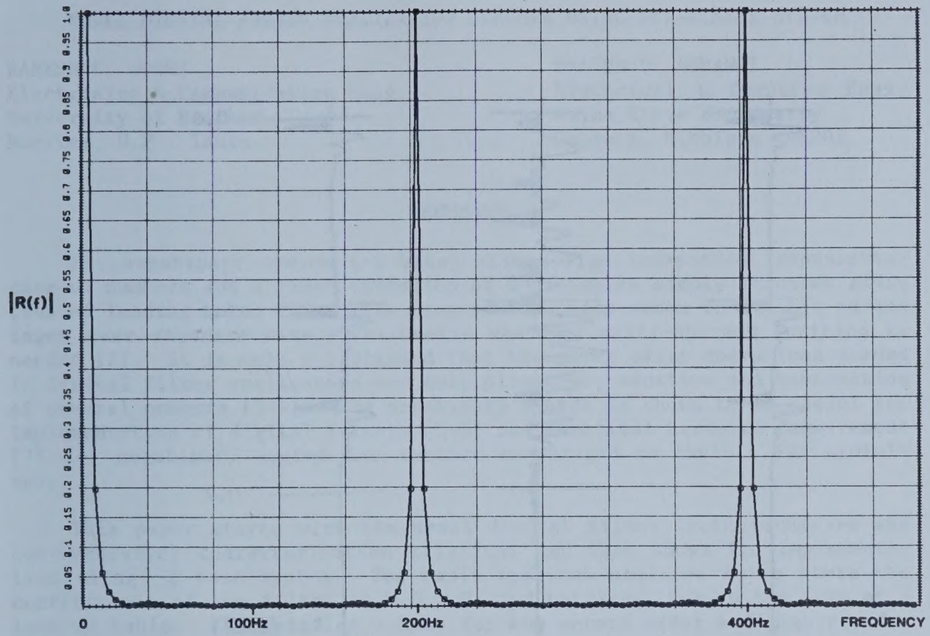


Figure 4: Optimum Filters Reject Response

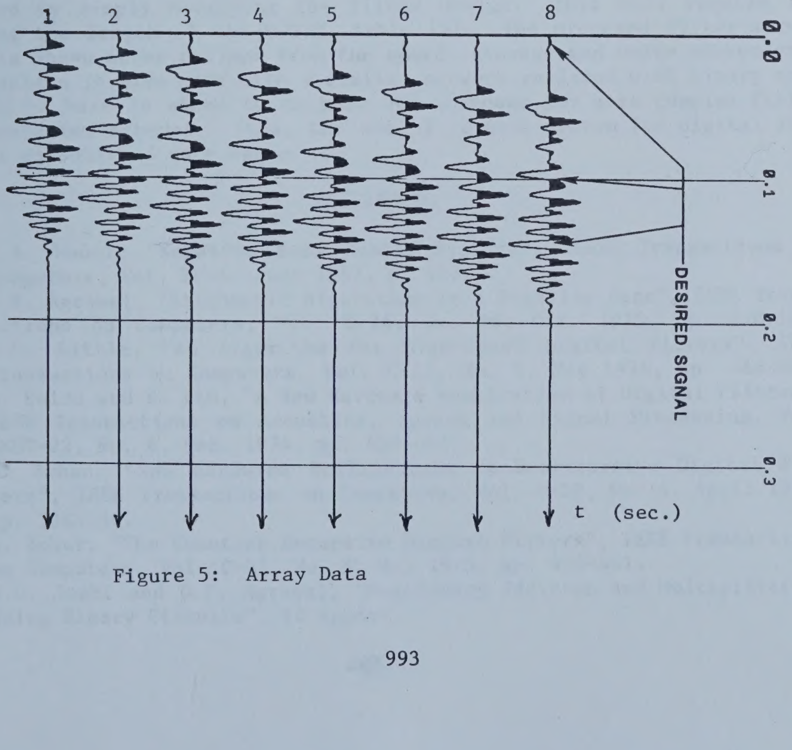


Figure 5: Array Data

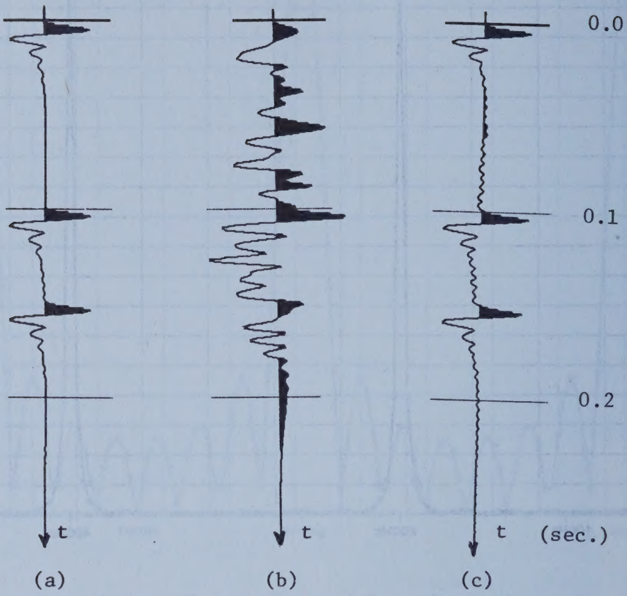


Figure 6: (a) Desired Signal, (b) Output of Delay-and-Sum Method, (c) Output of Optimum Filters Method.

NOVEL DIGITAL FILTER REALIZATION SCHEMES USING NEGABINARY SYSTEM

RAMESH C. JOSHI
Electronics & Communication Engg.
University of Roorkee
Roorkee, U.P., India

DHARMA P. AGRAWAL
Electrical & Computer Engg.
Wayne State University
Detroit, Michigan 48202

SUMMARY

The negabinary system (-2 base) allows sign independent representation of numbers and a range extension of the numbers simply requires addition of leading zeros [1]. This property has been shown to exhibit advantages over standard binary arithmetic whenever multi-operand addition is needed [2]. It is well established that the three basic operations needed in digital filter realization are multiplication, addition and subtraction of several numbers [3-4]. The negabinary system is shown to be useful for implementation of digital filters [5-6] and identical hardware requirement [7] for negabinary system has further encouraged to exploit its usefulness.

This paper starts with the usual digital filter (both recursive and non-recursive) characterization relations and then shows its implementation using -2 base system. The basic approach utilized is to store the coefficients of the filter in a R.O.M. and retrieve them in the form of a look-up table. The detailed scheme for the second order and higher order filter is shown and various digital circuits needed for its realization are also identified. To make the paper self-contained, multioperand negabinary adder and negabinary multiplier design [7] are illustrated in detail.

A wide range of filter forms and transfer functions could be easily obtained by simply modifying the filter design. This will require increasing the length of the R.O.M. table [4]. The proposed filter structure, is shown to be optimum from the speed, storage and noise viewpoints. A comparison is also made with a similar network realized with binary system and -2 base is shown to be more advantageous for more complex filter implementation schemes. Thus, the use of -2 base system for digital filters is established once again.

REFERENCES

1. L.B. Wadel, "Negative Base Number Systems", IEEE Transactions on Computers, Vol. EC-6, June 1957, p. 123.
2. D.P. Agrawal, "Arithmetic Algorithms in a Negative Base", IEEE Transactions on Computers, Vol. C-24, No. 10, Oct. 1975, pp. 120-125.
3. W.D. Little, "An Algorithm for High-Speed Digital Filters", IEEE Transactions on Computers, Vol. C-23, No. 5, May 1974, pp. 466-469.
4. A. Peled and B. Lin, "A New Hardware Realization of Digital Filters", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASST-22, No. 6, Dec. 1974, pp. 456-461.
5. S. Zohar, "New Hardware Realizations of Nonrecursive Digital Filters", IEEE Transactions on Computers, Vol. C-22, No. 4, April 1973, pp. 328-338.
6. S. Zohar, "The Counting Recursive Digital Filters", IEEE Transactions on Computers, Vol. C-22, No. 5, May 1973, pp. 433-441.
7. R.C. Joshi and D.P. Agrawal, "Negabinary Addition and Multiplication Using Binary Circuits", to appear.

A HARDWARE EFFICIENT ADM TO COMPANDED PCM CONVERTER USING A WINDOWED STEP RESPONSE DESIGN

B.W. JONES and J.L. LoCICERO
Illinois Institute of Technology
Department of Electrical Engineering
Chicago, Illinois 60616

SUMMARY

A hardware efficient system for converting from an adaptive delta modulation (ADM) bit stream to companded pulse code modulation (PCM) format has been devised. Emphasis on an improved decimation filter has lead to the formulation of a windowed unit step response design of a "G-filter". The companding, approximating the 8-bit μ-255 law, is performed with a digital system that is slightly less complex than the circuit devised by Kaneko [1].

The converter studied consists of an ADM step size generator, a decimation "G-filter" and a digital compressor. As LoCicero and Schilling [2] have shown, such a system (originally introduced by Goodman [3]) represents a hardware simplification over other converter structures [4]. We have compiled transfer characteristics comparing the "G-filter" to the standard finite impulse response (FIR) decimation filter when the ADM and PCM rates are set at 32K Hz and 8K Hz respectively, 12 filter coefficients are employed, and 6 different window functions are applied.

The performance of the entire converter is obtained via computer simulation and a time domain analysis technique. We have generated a family of signal-to-noise ratio (SNR) curves, for an unfiltered output signal, as a function of input signal power over a 60 dB dynamic range using a single-tone (800 Hz) sineusoid as the test signal. These results show that good performance is maintained if only 6 filter coefficients are retained and their accuracy greatly reduced.

REFERENCES

[1] H. Kaneko, "A Unified Formulation of Segment Companding Laws and Synthesis of Codecs and Digital Companders," B.S.T.J., vol. 49, pp. 1555-1588, Sept. 1970.

[2] J.L. LoCicero and D.L. Schilling, "An All-Digital Technique for ADM to PCM Conversion," in Proc. IEEE Nat. Telecommun. Conf., pp. 29.3-1 to 29.3-6, Dec. 1976.

[3] D.J. Goodman, "The Application of Delta Modulation to Analog-to-Digital Encoding," B.S.T.J., vol. 48, pp. 321-343, Feb. 1969.

[4] H. Jeong and C.K. Un, "A PCM/ADM Code Converter," IEEE Trans. Acoust., Speech, Sig. Proc., vol. ASSP-27, pp. 762-768, Dec. 1979.

A PROOF OF LOCALIZED STEADY-STATE RESPONSE IN LARGE INTERCONNECTED POWER SYSTEM

Natarajan Narasimhamurthi
Department of Electrical
and Computer Engineering
University of Michigan,
Ann Arbor, MI 48104

Marija Ilić-Spong
Department of Systems Science
and Mathematics
Washington University
St. Louis, MO 63130

0. ABSTRACT

When analyzing the effect of a fault (disturbance) in a large interconnected power system, it is either explicitly or implicitly assumed that the effect of the fault is more pronounced close to the location of the fault and that the effect diminishes as one moves further away from the location of the fault. This paper provides an elementary proof of this phenomenon, with the power system modelled by the simplified, nonlinear, 'real power flow - phase angle model'.

1. INTRODUCTION

When analyzing the effect of a fault (disturbance) in a large interconnected power system it is tacitly assumed that the effect of the fault becomes progressively smaller as one moves away from the location of the fault. This intuitively appealing assumption is the basis for replacing the part of the system far from the location of the fault by an approximate model, such as a linearized model or even an approximation of it. Zaborszky et.al. have utilized this in developing their fast contingency evaluation method using 'Concentric Relaxation' which rests on the assumption that buses far from the fault do not undergo any change in their complex voltages [1].

In this paper the simplified nonlinear P-θ model, is considered and an elementary proof of the proposition which uses only the following properties of the node admittance matrix, is presented:

In the case of the node-admittance matrix of a resistive network with positive resistance.

a) The sum of the entries on any row of the matrix is less than or equal to zero

b) All the elements of the inverse of the matrix are non negative

c) Even if any number of rows and corresponding columns are deleted, the properties a) and b) still hold.

These properties can be easily deduced from the algorithms by which these matrices are constructed [2].

2. MODEL

2.1 Assumptions

In modelling the power system, it is assumed that

a) The transmission lines are lossless and inductive

b) The magnitudes of the voltages at the buses are constant

c) There is a reference bus (refered to as the slack bus) where the phase angle of the voltage is also constant

d) The postfault injections are modelled as a change in the real power injection at a pair of buses (one of them could be the slack bus) and the effect of the fault is the steady state change in the voltage phase angle at the buses

e) Prior to the fault, the system is in a steady state with the angles across each transmission line $< \frac{\pi}{2}$

2.2 Definitions

It is assumed that the lines in the system are given an orientation a priori. Let

p Vector of real power flows in the transmission lines

ψ Vector of phase angle difference across transmission lines

P Vector of real power injections at the buses

θ Vector of voltage phase angles at the buses

A Reduced bus incidence matrix [1,2]

$1] = [1 \quad 1 \quad 1 \quad \ldots \quad 1]^T$

$p^o, \psi^o, P^o, \theta^o$ Respective values of the quantities before the fault

$\Delta p, \Delta\psi, \Delta P, \Delta\theta$ Respective changes in the quantities after the fault

2.3 Equations

Network Equations

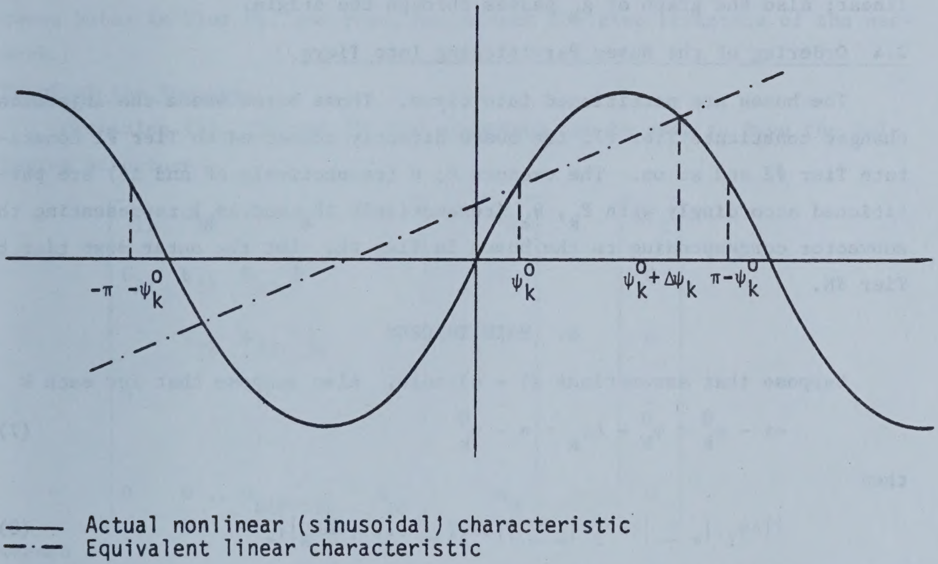
$$Ap = P \tag{1}$$

$$A^T\theta = \psi \tag{2}$$

Transmission Line Characteristics

In view of assumptions a) and b) it can be easily derived [3] that for the line k going from bus i to bus j

$$p_k = \frac{E_i E_j}{x_k} \sin \psi_k = F_k \sin \psi_k \tag{3}$$



-π
$-\psi_k^0$
ψ_k^0
$\psi_k^0+\Delta\psi_k$
$\pi-\psi_k^0$
—— Actual nonlinear (sinusoidal) characteristic
—·— Equivalent linear characteristic

Figure 1

where E_i and E_j are the magnitudes of the voltages at buses i and j and x_k is the inductance of the line. From (1), (2) and (3), it is seen that

$$A \, \Delta p = \Delta P \tag{4}$$

$$A^T \, \Delta\theta = \Delta\psi \tag{5}$$

$$\begin{aligned}\Delta P_k &= F_k \sin(\psi_k^0 + \Delta\psi_k) - \sin \psi_k^0 \\ &= g_k \, (\Delta\psi_k)\end{aligned} \tag{6}$$

It should be noted that (4) and (5) are linear equations while (6) is nonlinear; also the graph of g_k passes through the origin.

2.4 Ordering of the Buses Partitioning Into Tiers

The buses are partitioned into tiers. Those buses where the injection changes constitute Tier #1, the buses directly connected to Tier #1 constitute Tier #2 and so on. The vectors P, θ (respectively ΔP and Δθ) are partitioned accordingly with P_k, θ_k (respectively ΔP_k and $\Delta\theta_k$) representing the subvector corresponding to the buses in Tier #k. Let the outer most tier be Tier #N.

3. MAIN THEOREM

Suppose that assumptions a) - e) hold. Also suppose that for each k

$$-\pi - \psi_k^0 < \psi_k^0 + \Delta\psi_k < \pi - \psi_k^0 \tag{7}$$

then

$$||\Delta\theta_1||_\infty \geq ||\Delta\theta_2||_\infty \geq ||\Delta\theta_3||_\infty \; \ldots \; ||\Delta\theta_N||_\infty \tag{8}$$

Remarks: (i) (7) defines an interval of length 2π which always includes $(\frac{-\pi}{2}, \frac{\pi}{2})$, the interval normally of interest.

(ii) Since (8) is in terms of ∞-norms, it states that the largest change in Tier #1 is not less then the largest change in Tier #2 etc.

Proof: Note that it is enough to show that

$$||\Delta\theta_{k-1}||_\infty \geq ||\Delta\theta_k||_\infty \qquad k = 2,3...N \tag{9}$$

The proof utilizes the following two Lemmas.

Lemma 1: If the theorem is true when equation (6) is replaced by the following linear equation

$$\Delta p_k = G_k \, \Delta\psi_k; \; G_k > 0 \tag{10}$$

then the theorem is true for the nonlinear equations also.

Proof: In view of (7) the nonlinear problem (6) can be replaced by a linear problem (9) as illustrated in Fig. 1. As the linear problem has a unique solution the two solutions must coincide.

Remark: Lemma 1 is the crux of the whole proof which transforms the nonlinear problem into a familiar linear problem and uses only the property that the graph of $g_k(\cdot)$ lies in the first and third quadrant. Thus, one only needs results from linear circuit theory.

Lemma 2: In the case of the linear problem it is enough to prove (8) for $k = N$.

Proof: For $k < N$, one can by star-mesh transformations (Ward-reduction) eliminate the nodes in Tier #k+1 through #N without affecting the solution in Tiers #1 through #k. Also, the reduction will introduce new lines between buses in Tier #k, and thus, not affect the tier structure of the network.

Proof of the Theorem:

Combining (4), (5) and (9) the equations can be seen to have the following structure

$$\begin{bmatrix} G_{11} & G_{12} & 0 & 0 & 0 \\ G_{21} & G_{22} & G_{23} & 0 & \\ & G_{32} & G_{33} & G_{34} & \\ 0 & & & & \\ 0 & 0 \,.. & G_{N(N-1)} & & G_{NN} \end{bmatrix} \begin{bmatrix} \Delta\theta_1 \\ \Delta\theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \Delta\theta_N \end{bmatrix} = \begin{bmatrix} \Delta P_1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

whence

$$G_{N(N-1)}\Delta\theta_{N-1} + G_{NN}\Delta\theta_N = 0$$

or

$$\Delta\theta_N = (-G_{NN}^{-1}\, G_{N(N-1)})\,\Delta\theta_{N-1} \tag{11}$$

Hence

$$||\Delta\theta_N||_\infty \leq ||-G_{NN}^{-1}\, G_{N(N-1)}||_\infty \; ||\Delta\theta_{N-1}||_\infty \tag{12}$$

where

$$||-G_{NN}^{-1}\, G_{N(N-1)}||_\infty = \max_i \sum_j |(-G_{NN}^{-1}\, G_{N(N-1)})_{i,j}| \triangleq \nu \tag{13}$$

From the elementary properties of the admittance matrix every element of G_{NN}^{-1} is non negative and every element of $G_{N(N-1)}$ is non positive. Hence,

$$\nu = \max_i \sum_j (-G_{NN}^{-1}\, G_{N(N-1)})_{i,j} = \max_i (-G_{NN}^{-1}\, G_{N(N-1)}1])_i \tag{14}$$

Since the sum of entries in each row is greater than or equal to zero

$$G_{NN}\, 1] + G_{N(N-1)}\, 1] \geq 0 \tag{15}$$

where $\leq$ means component-wise in equality. Multiplying (14) on both sides by G_{NN}^{-1}, every entry of which is non negative,

$$1] + G_{NN}^{-1}\, G_{N(N-1)}\, 1] \geq 0$$

or

$$-G_{NN}^{-1}\, G_{N(N-1)}\, 1] \leq 1], \text{ i.e. } \nu \leq 1$$

which proves the Theorem.

4. CONCLUSION

The importance of this paper rests not only on the fact that it provides the only known proof for the localized steady state response but also on the method used to prove the result. In view of the general nature of the proof (it only requires the function $g_k(\cdot)$ in (6) to be in the first and third quadrant) it is felt that the result can be proved for a more elaborate model which includes line losses as well. Also, Lemma 1 which transforms the nonlinear problem into a linear problem is new in this context, though identical transformation has been used to prove the 'maximum voltage gain theorem' in circuit theory [4]. In fact the result in this paper can be viewed as a generalization of the maximum voltage gain theorem.

REFERENCES

1. J. Zaborszky, K. W. Whang, and K. V. Prasad, "Fast Contingency Evaluation Using Concentric Relaxation," IEEE PAS Transactions, January/February 1980, pp. 28-36.

2. G. W. Stagg and A. H. El-Abiad, Computer Methods in Power System Analysis, McGraw Hill, New York, 1968.

3. O. I. Elgerd, Electric Energy Systems Theory: An Introduction, McGraw Hill, 1971.

4. Desoer and Kuh, Basic Circuit Theory, McGraw Hill, 1969.

INDEX OF AUTHORS

Abidi, M. A. 822
Adams, G. B., III 544
Agarwal, D. P. 995
Akl, S. G. 250
Alden, R. T. H. 285
Alexander, W. E. 986
Anderson, J. B. 68
Arce, G. R. 945
Ashok, V. 643
Avis, D. M. 35
Azagoury, M. 633
Balducchi, J.-F. 891
Barmish, B. R. 697
Becker, A. 618
Bentley, J. L. 11, 41
Berger, T. 433
Bertolazzi, P. 660
Bertsekas, D. P. 584
Bhattacharyya, S. P. 104, 105
Blahut, R. E. 50
Briggs, F. A. 554
Brown, M. E. 534
Bucklew, J. A. 454
Burns, J. E. 833
Calvaer, A. 735
Caroubalos, C. 88
Carruthers, W. 11
Castor, K. G. 976
Chaćon, A. 633
Chang, H. 267
Chang, M. U. 433
Chellappa, R. 956
Chen, K.-W. 564
Cheno, L. 223
Chung, C.-G. 124
Chyung, D. H. 98
Cohen, G. 891
Cooper, G. R. 78
Dallas, S. L. 824
Davison, E. J 91
DeCarlo, R. 909
DeClaris, N. 594
Doyle, J. C. 715
Dubois, M. 554
Dudek, K. P. 386
Dunham, J. G. 438, 440
El-Nahas, I. 285
El-Sakkary, A. 380
Ephremides, A. 339
Eslami, M. 424
Feather, A. E. 985
Fernandez, B. 633
Fiduccia, C. M. 812
Flajolet, P. 223
Francon, J. 223
Friedlander, B. 574
Fuchs, H. 677
Gallagher, N.C.,Jr. 446,926,945
Gheury, J. 735
Gibson, J. D. 955,966
Goodwin, G. C. 623
Gowda, I. G. 1
Gray, R. M. 437
Hakimi, S. L. 480,824
Halverson, D. R. 765
Han, K.-W. 124
Hart, J. M. 29
Harvey, C. A. 715
Hilson, D. W. 196
Hirschberg, D. S. 823
Hogan, D. J. 260
Hollot, C. V. 697
Hüseyin, Ö. 900
Ikeda, M. 726
Ilic, D. D. 792
Ilic-Spong, M. 997

Irani, K. B. 564
Jaffe, J. M. 668
Ja'Ja', J. 643
Javid, S. H. 881
Johnson, C. R. 623
Johnson, R. R. 196
Jones, B. W. 996
Joshi, R. C. 995
Kajitani, Y. 507
Kalouptsidis, N. 88
Kashyap, R. L. 956
Kassam, S. A. 755,774
Kawakami, T. 507
Kedem, Z. M. 677
Kempner, K. M. 594
Khalil, H. K. 405
Kieffer, J. C. 438
Kim, H. Y. 106
Kim, J. 98
Kirkpatrick, D. G. 1, 243
Korach, E. 687
Kuhlmann, F. 935
Kumar, P. R. 614,618
Lavi, A. 145
Lazar, A. A. 784
Lee, C. C. 745
Lee, D. T. 267
Lee, H. 341
Lefebvre, S. 909
Leiss, E. 490,491
Leung, J. Y.-T. 464,471,480
Lewis, F. D. 490
Liebetreu, J. M. 351
LoCicero,J. L. 996
Lucertini, M. 660
Lynch, N. A. 833
Manolakis, D. 88
Marks, G. E. 134
Martella, G. 843
Matis, K. R. 60
Matsuo, T. 303
Mattheyses, R. M. 812
McClamroch, N. H. 919
McClure, J. W. 707
McGillem, C. D. 801
McGree, T. P. 78
McMillen, R. J. 544
McVay, J. W. 966
Meditch, J. S. 333
Meijer, H. 250
Metzner, J. J. 822
Milito, R. A. 860
Modestino, J. W. 60
Monien, B. 650
Morf, M. 574
Moustakides, G. 774
Murata, T. 497
Nakajima, K. 480
Narasimhamurthi, N. 997
Narayan, P. 89
Neuhoff, D. L. 359,976
Nodes, T. A. 926
Oliveira, V. A. 104
Olson, D. E. 986
Padilla, C. S. 633,860
Padilla, R. A. 633,860
Papavassilopoulos, G. P. 851
Pelton, F. 177
Pomalaza, C. A. 801
Postlethwaite, I. 396
Puech, C. 223
Quadrat, J. P. 870
Reininger, R. C. 966
Ribbens-Pavella, M. 735
Richter, S. 909
Rines, K. D. 446
Robertazzi, T. G. 946
Rotem, D. 687

Rowbotham, T. R. 369
Ryan, C. R. 351
Sabatel, J. 735
Sadanandan, N. D. 196
Sain, M. K. 386,604
Santoro, N. 687
Saxe, J. B. 41
Schafer, R. M. 386
Schaub, P. E. 167,177
Schreiber, F. 843
Schwartz, S. C. 784,946
Sendaula, H. M. 196
Sennott, J. W. 340
Sennott, L. I. 340
Sezer, M. E. 900
Shanthikumar, J. G. 524
Shapiro, E. Y. 106
Sherrod, E. E. 986
Shin, J. G. 774
Shippey, E. D. 167
Siegel, H. J. 544
Siegel, L. J. 985
Siljak, D. D. 726,900
Simaan, M. 987
Sin, K. S. 623
Sinha, N. K. 285
Skelton, R. E. 323,414
Sloane, N. J. A. 67
Snyder, D. L. 89
Soloway, S. R. 514
Spaccamela, A. M. 660
Srivatsa, K. 68
Sudborough, I. H. 650
Sullivan, R. L. 214
Suzuki, I. 497
Suzuki, M. 595
Thomas, G. 155
Thomas, J. B. 745
Thomas, R. J. 206,707
Thorp, J. S. 206,707
Toussaint, G. T. 20
Tugnait, J. K. 313
Tung-Kleinberg, S. Y. 433
Valiant, L. G. 802
Varshney, P. K. 379
Viot, M. 870
Vuillemin, J. 223
Wall, J. E., Jr. 715
Wang, J. 275
Wang, S.-H. 91
Wasynczuk, O. 187
Weide, B. W. 260,534
Whitehead, J. 464
Wise, G. L. 454,765,935
Wong, C. K. 267
Wormley, D. M. 155
Wright, W. E. 233
Wu, K.-T. 359
Wu, W. W. 369
Xia, G.-H. 295
Yedavalli, R. K. 414
Yeh, H.-H. 124
Young, K. D. 116
Yurkovich, S. 604
Zames, G. 380

OUND BY

UNIVERSITY OF ILLINOIS-URBANA
621.3811AL5 C00A
PROCEEDINGS / ALLERTON CONFERENCE ON COM
1980
3 0112 008149111